



# Ordonnancement de rendez-vous en tête à tête

Agnès Le Roux

## ► To cite this version:

Agnès Le Roux. Ordonnancement de rendez-vous en tête à tête. Automatique. Ecole des Mines de Nantes, 2014. Français. NNT : 2014EMNA0182 . tel-01127096

**HAL Id: tel-01127096**

**<https://theses.hal.science/tel-01127096>**

Submitted on 7 Mar 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Thèse de Doctorat

**Agnès LE ROUX**

*Mémoire présenté en vue de l'obtention du*  
**grade de Docteur de l'École nationale supérieure des mines de Nantes**  
*sous le label de l'Université de Nantes Angers Le Mans*

**École doctorale : Sciences et technologies de l'information, et mathématiques**

**Discipline : Informatique et applications, section CNU 27**

**Spécialité : Recherche opérationnelle**

**Unité de recherche : Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)**

**Soutenue le 24 Octobre 2014**

**Thèse n° : 2014EMNA0182**

## Ordonnancement de rendez-vous en tête-à-tête

### JURY

Rapporteurs : **M. Jean Charles BILLAUT**, Professeur des Universités, École Polytechnique de l'Université de Tours  
**M. Jacques CARLIER**, Professeur des Universités, Université de Technologie de Compiègne

Examineurs : **M. Christian ARTIGUES**, Directeur de recherche, LAAS-CNRS  
**M. Éric HUMLER**, Directeur de la recherche, École nationale supérieure des Mines de Nantes  
**M. Narendra JUSSIEN**, Professeur de l'institut Mines-Telecom, Directeur de Telecom Lille, Telecom Lille

Directrice de thèse : **M<sup>me</sup> Christelle GUÉRET**, Professeur des Universités, Université d'Angers

Co-encadrante : **M<sup>me</sup> Odile BELLENGUEZ-MORINEAU**, Chargée de recherche, École nationale supérieure des Mines de Nantes, IRCCyN



# Remerciements

Je remercie tout d'abord mes directrices de thèse Christelle GUÉRET et Odile MORINEAU qui m'ont donné la chance d'entreprendre cette thèse de doctorat, m'ont suivi et encouragé tout au long de ce travail et pour les nombreuses relectures et corrections du manuscrit.

Je remercie grandement Jean-Charles BILLAUT et Jacques CARLIER qui m'ont fait l'honneur d'être rapporteurs de cette thèse ainsi que Christian ARTIGUES, Éric HUMLER et Narendra JUSSIEN pour avoir été membres de mon jury et s'être intéressés de près à mon travail.

Je remercie chaleureusement l'ensemble de mes collègues qui m'ont accueilli au sein du département Automatique et Productique de l'École des Mines de Nantes et avec lesquels j'ai eu de nombreux échanges enrichissants. Notamment, je remercie tous les doctorants qui m'ont apporté une part de leur expérience sur les joies et les peines de la thèse. Mes excellents co-bureaux Majid ESKANDARPOUR et Tanguy LAPÈGUE ont été ceux que j'ai le plus souvent sollicités et je les remercie pour toutes nos discussions sur des sujets plus ou moins scientifiques.

Je remercie également Damien PROT avec qui j'ai eu plaisir de travailler sur l'écriture d'un article.

Je remercie toute ma famille pour le soutien et les encouragements qu'ils m'ont apportés dans ce projet. Je remercie mon compagnon Michaël BALAN pour sa confiance et son soutien indéfectibles.

Enfin, à tous les aikidokas croisés sur les tatamis, en particulier mes professeurs Philippe OUJAGIR et Farid SI MOUSSA : dōmo arigatōgozaimasu pour tout ce que vous m'avez appris durant ma dernière année de thèse.



# Liste des notations

$\mathcal{M} = \{M_0, M_1, \dots, M_{m-1}\}$	Ensemble des hommes
$\mathcal{F} = \{F_0, F_1, \dots, F_{n-1}\}$	Ensemble des femmes
$m = \text{card } \mathcal{M}$	Nombre d'hommes
$n = \text{card } \mathcal{F}$	Nombre de femmes
$\mathcal{F}_{M_i}$	Ensemble des femmes que rencontre $M_i$
$\mathcal{M}_{F_j}$	Ensemble des hommes que rencontre $F_j$
$\mathcal{S} = \llbracket 0, s-1 \rrbracket$	Ensemble des créneaux de l'événement
$P = (p_{ij})_{i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket}$	Matrice des rencontres ( $p_{ij} = 1$ ssi l'homme $M_i$ rencontre la femme $F_j$ )
$\text{Meet} = \{(i, j) \in \mathcal{M} \times \mathcal{F} \mid p_{ij} = 1\}$	Ensemble des rencontres à affecter
$(U \cup V, E)$	Graphe des rencontres associé à la matrice $P$
$r_i^{(1)}$	Date d'arrivée ou de disponibilité de l'homme $M_i$
$r_j^{(2)}$	Date d'arrivée ou de disponibilité de la femme $F_j$
$r_{ij} = \max(r_i^{(1)}, r_j^{(2)})$	Date de début au plus tôt de la rencontre entre l'homme $M_i$ et la femme $F_j$
$r_{\max}$	Dernière date d'arrivée des participants
$e_i^{(1)}$	Date de fin au plus tôt de l'homme $M_i$
$e_j^{(2)}$	Date de fin au plus tôt de la femme $F_j$
$d_{ij} = \min(e_i^{(1)}, e_j^{(2)})$	Date d'échéance de la rencontre entre l'homme $M_i$ et la femme $F_j$
$d_{\min}$	Plus petite date d'échéance des rencontres
$d_{\max}$	Plus grande date d'échéance des rencontres
$C_{ij}$	Date de fin de la rencontre entre l'homme $M_i$ et la femme $F_j$
$t_{ij} = C_{ij} - 1$	Date de début de la rencontre entre l'homme $M_i$ et la femme $F_j$
$C_i^{(1)}$	Date de fin des rencontres de l'homme $M_i$
$C_j^{(2)}$	Date de fin des rencontres de la femme $F_j$
$W_i^{(1)}$	Nombre de créneaux d'attente de l'homme $M_i$
$W_j^{(2)}$	Nombre de créneaux d'attente de la femme $F_j$
$W_{\max}^{(1)}$	Nombre maximal de créneaux d'attente des hommes
$W_{\max}^{(2)}$	Nombre maximal de créneaux d'attente des femmes
$W_{\max}$	Nombre maximal de créneaux d'attente des participants



# Table des matières

<b>I</b>	<b>Présentation du problème et état de l'art</b>	<b>13</b>
<b>1</b>	<b>Présentation du problème</b>	<b>15</b>
1.1	Définition des problèmes d'ordonnancement de rendez-vous en tête-à-tête	15
1.2	Caractéristiques du problème de speed-dating	16
1.3	Notations	16
1.3.1	Données	16
1.3.2	Variables	17
1.3.3	Notations à trois champs $\alpha \beta \gamma$	17
1.3.4	Représentation d'une solution	18
1.4	Génération d'instances	18
1.5	Conclusion	19
<b>2</b>	<b>État de l'art</b>	<b>21</b>
2.1	Introduction	21
2.2	Problèmes d'ordonnancement de rendez-vous en tête-à-tête	21
2.3	Autres problèmes de rencontres	23
2.4	Problèmes proches du problème de speed-dating	24
2.4.1	Problème d'open shop	24
2.4.2	Problème d'emploi du temps	25
2.4.3	Problèmes d'ordonnancement par fournées avec incompatibilités entre les jobs	25
2.4.4	Coloration d'arêtes d'un graphe par intervalles	26
2.4.5	Problème de multiflot entier	26
2.5	Conclusion	27
<b>II</b>	<b>Résultats de complexité</b>	<b>29</b>
<b>3</b>	<b>Résultats généraux de complexité</b>	<b>31</b>
3.1	Introduction	31
3.2	Réductions évidentes	32
3.2.1	Réductions dans le cas général	32
3.2.2	Réductions dans le cas où des rencontres sont interdites	32
3.2.3	Rôles symétriques des femmes et des hommes	32
3.3	Conclusion	33
<b>4</b>	<b>Problèmes polynomiaux</b>	<b>35</b>
4.1	Introduction	35
4.2	Cas sans rencontres interdites et sans retards des participants	35
4.3	Cas sans rencontres interdites avec retards pour la population la moins nombreuse	36
4.4	Cas sans rencontres interdites et avec retards pour une population	37
4.5	Cas particulier : minimisation de la date de fin globale	41



<b>5</b>	<b>Problèmes <math>\mathcal{NP}</math>-difficiles</b>	<b>43</b>
5.1	Introduction	43
5.2	Cas avec rencontres interdites et retards autorisés	43
5.3	Cas avec rencontres interdites et retards pour une population	46
5.4	Cas avec rencontres interdites et sans retards	48
5.5	$\mathcal{NP}$ -difficulté au sens fort	51
5.6	Non approximabilité	52
<b>III</b>	<b>Bornes inférieures et bornes supérieures</b>	<b>55</b>
<b>6</b>	<b>Bornes inférieures</b>	<b>57</b>
6.1	Introduction	57
6.2	Borne inférieure dépendant des dates d'échéance des rencontres	58
6.3	Borne inférieure dépendant des dates d'arrivée des participants	59
6.4	Borne inférieure issue de couplages maximums	62
6.5	Condition nécessaire de non attente des participants	64
6.6	Résultats expérimentaux	65
6.7	Conclusion	66
<b>7</b>	<b>Bornes supérieures</b>	<b>69</b>
7.1	Introduction	69
7.2	Bornes déterminées par des heuristiques d'ordonnancement en série	70
7.3	Bornes déterminés par des heuristiques d'ordonnancement en parallèle	71
7.4	Résultats expérimentaux	71
7.4.1	Bornes déterminées par des heuristiques d'ordonnancement en série	72
7.4.2	Bornes déterminées par heuristiques d'ordonnancement en parallèle	74
7.4.3	Comparaison bornes inférieures/bornes supérieures	75
7.5	Conclusion	75
<b>IV</b>	<b>Modélisation et résolution</b>	<b>77</b>
<b>8</b>	<b>Programmation linéaire en nombres entiers</b>	<b>79</b>
8.1	Introduction	79
8.2	Modèle indexé sur le temps	79
8.2.1	Modèle de base	80
8.2.2	Inégalités valides pour le modèle indexé sur le temps	81
8.3	Extension du modèle indexé sur le temps	81
8.4	Modèle basé sur les dates de rencontres	83
8.4.1	Modèle de base	83
8.4.2	Inégalités valides	84
8.5	Règles de dominance	84
8.5.1	Rencontres à la date zéro	84
8.5.2	Rencontres aux dates $r_{ij}$ ( $Dr_{ij}$ )	86
8.5.3	Ordonnements actif et semi-actif (AS et SAS)	88
8.5.4	Symétries (Sym)	89
8.5.5	Interdiction de créneaux vides (NES) (No Empty Slot)	92
8.6	Résultats expérimentaux	93
8.6.1	Relaxation continue	93
8.6.2	Temps de résolution	94
8.7	Considérations sur la génération de colonnes	97
8.8	Conclusion	98

<b>9</b>	<b>Programmation par contraintes</b>	<b>99</b>
9.1	Introduction	99
9.2	Modèle et stratégies de recherche	100
9.3	Réduction du domaine de recherche	101
9.3.1	Ordonnancement actif	101
9.3.2	Symétries des participants	101
9.3.3	Symétries des dates de rencontres	101
9.4	Résultats expérimentaux	103
9.4.1	Stratégies de recherche	103
9.4.2	Règles de dominance	104
9.5	Conclusion	105
<b>10</b>	<b>Recherche à divergence limitée</b>	<b>107</b>
10.1	Introduction	107
10.2	Schéma de branchement et critère heuristique	108
10.3	Élagage de l'arbre	109
10.3.1	Ordonnancement maximale-actif	109
10.3.2	Bornes inférieures et supérieures	110
10.3.3	Gestion des symétries	111
10.4	Expérimentations numériques	113
10.4.1	Performance des coupes	113
10.4.2	Version avec coupes	115
10.4.3	Fréquence de calcul d'une borne supérieure	116
10.5	Conclusion	117
	<b>Appendices</b>	<b>121</b>
<b>A</b>	<b>Modèles de génération de colonnes</b>	<b>123</b>
A.1	Modèle indicé par les dates de rencontres	123
A.1.1	Problème maître	123
A.1.2	Sous-problèmes	124
A.2	Modèle indicé par les participantes	124
A.2.1	Problème maître	124
A.2.2	Sous-problèmes	125



# Introduction

Parmi les thématiques d'intérêt majeur en optimisation, figurent les problèmes d'ordonnancement qui répondent à des besoins essentiels d'organisation et de réduction des coûts pour les entreprises. De manière générale, les problèmes d'ordonnancement traitent de l'attribution de dates d'exécution à des tâches nécessitant des ressources limitées sous des contraintes spécifiques. Le but est de positionner l'ensemble de ces tâches de manière à leur allouer les ressources nécessaires et respecter toutes les contraintes en optimisant un (ou des) objectif(s) fixé(s). Les applications sont nombreuses notamment dans les domaines de l'industrie, de l'informatique ou la gestion de projets.

Assez récemment, un nouveau type de problèmes d'ordonnancement est apparu qui a pour but de planifier des rendez-vous en tête-à-tête. Il s'agit d'organiser en parallèle des rendez-vous de courte durée prédéterminée réunissant deux participants. Ce genre de problème se rencontre dans diverses applications telles que des forums pour l'emploi où des recruteurs rencontrent des demandeurs d'emploi, des salons professionnels organisant des rendez-vous d'affaires, des réunions parents-professeurs ou des soirées de speed-dating. La planification de tels événements à la main est souvent une tâche longue et fastidieuse en raison du grand nombre de participants et des nombreuses contraintes à respecter. Il faut par ailleurs chercher à satisfaire les participants au mieux dans le nombre et le choix des rendez-vous planifiés car les rencontres peuvent avoir de gros enjeux pour les participants (signature de contrats, recherche de partenariats, vente de produits, recherche de travail).

Certaines applications de problèmes d'ordonnancement de rendez-vous en tête-à-tête ont fait l'objet d'études dans la littérature de manière isolée. À notre connaissance, ces problèmes n'ont pas été regroupés et classifiés de manière globale. Dans cette thèse, nous référençons différentes applications de ce type de problèmes et étudions un cas particulier qui n'a pas encore été traité.

Cette thèse est constituée de quatre parties.

La partie [I](#) de ce document est consacrée à la description des problèmes d'ordonnancement de rendez-vous en tête-à-tête et à l'état de l'art de ces problèmes. Cette partie est divisée en deux chapitres. Dans le chapitre [1](#), nous définissons formellement les problèmes d'ordonnancement de rendez-vous en tête-à-tête et explicitons les spécificités du problème de speed-dating. Nous proposons des notations pour ces problèmes étendant les notations classiques  $\alpha|\beta|\gamma$ . Enfin, nous détaillons les paramètres utilisés pour la génération d'instances de tests. Dans le chapitre [2](#), nous présentons des applications de problèmes d'ordonnancement de rendez-vous étudiées dans la littérature ainsi que d'autres problèmes classiques pouvant se rapprocher des problèmes de speed-dating. Pour chacun d'entre eux, nous décrivons leurs caractéristiques ainsi que leur complexité lorsqu'elle est connue.

Dans la partie [II](#), nous étudions la complexité des problèmes de speed-dating. Le chapitre [3](#) pose le cadre de ces résultats en énonçant des réductions évidentes selon les caractéristiques du problème. Dans le chapitre [4](#), nous démontrons la complexité de cas polynomiaux et dans le chapitre [5](#), nous démontrons que les cas avec rencontres interdites sont  $\mathcal{NP}$ -difficiles au sens fort. Ces démonstrations sont basées sur la réduction d'une généralisation  $\mathcal{NP}$ -complète du problème de coloration d'arêtes dans un graphe biparti.

Nous proposons ensuite des bornes inférieures et supérieures pour le problème de speed-dating dans la partie [III](#). Dans le chapitre [6](#), nous relâchons des contraintes de disjonctions et étudions les problèmes obtenus pour en déduire des bornes inférieures. Dans le chapitre [7](#), nous proposons plusieurs heuristiques gloutonnes basées sur l'ordonnance-

ment séquentiel des rencontres par dates. Nous nous appuyons sur des critères de priorité pour choisir les rencontres à effectuer en premier.

La partie IV comprend trois chapitres dédiés à la modélisation et la résolution du problème de speed-dating. Dans le chapitre 8, nous présentons deux modèles de programmation linéaire en nombres entiers pour notre problème. Dans un but de renforcement de ces modèles, nous leur ajoutons des inégalités valides et des règles de dominance. Dans le chapitre 9, nous présentons un modèle de programmation par contraintes de notre problème. Nous proposons plusieurs stratégies de recherche et ajoutons également des règles de dominance permettant de casser les symétries du problème. L'ensemble de ces modèles est testé sur un ensemble de 300 instances générées aléatoirement. Le chapitre 10 est consacré au développement d'une recherche à divergence limitée (Limited Discrepancy Search). La méthode consiste à explorer un arbre de recherche tronqué en suivant une stratégie d'exploration se basant sur un critère heuristique. Les heuristiques définies dans le chapitre 7 sont exploitées dans cette méthode pour définir la stratégie d'exploration. Afin de faire converger cette méthode plus rapidement vers une bonne solution, nous déterminons des conditions de coupes de branches basées en particulier sur la gestion de solutions symétriques. Cette méthode est également testée sur les 300 instances que nous avons générées.

Nous concluons ce manuscrit par la synthèse des travaux que nous avons effectués ainsi qu'une présentation de perspectives et recherches futures.

## **Première partie**

### **Présentation du problème et état de l'art**



# Chapitre 1

## Présentation du problème

### Sommaire

<b>1.1</b>	<b>Définition des problèmes d'ordonnancement de rendez-vous en tête-à-tête</b>	<b>15</b>
<b>1.2</b>	<b>Caractéristiques du problème de speed-dating</b>	<b>16</b>
<b>1.3</b>	<b>Notations</b>	<b>16</b>
<b>1.4</b>	<b>Génération d'instances</b>	<b>18</b>
<b>1.5</b>	<b>Conclusion</b>	<b>19</b>

Dans ce chapitre, nous définissons de manière formelle les problèmes d'ordonnancement de rendez-vous en tête-à-tête et le cas particulier du problème de speed-dating. Nous introduisons les notations utilisées tout au long de ce manuscrit et décrivons la génération des instances.

### 1.1 Définition des problèmes d'ordonnancement de rendez-vous en tête-à-tête

Dans un problème d'ordonnancement de rendez-vous en tête-à-tête, différents participants souhaitent se rencontrer au cours de rendez-vous de courte durée. Les contraintes sont les suivantes :

- deux participants se rencontrent au plus une fois ;
- une rencontre implique exactement deux participants ;
- deux participants ne peuvent pas faire plus d'une rencontre à la fois.

Les rencontres s'effectuent en parallèle et en général ont lieu au même endroit.

Selon le type de problème, plusieurs spécificités sont prises en compte. Les participants peuvent se scinder en deux populations distinctes. Dans ce cas, une rencontre ne se fait qu'entre deux participants de populations différentes. Dans le cas où la population est homogène, il n'y a pas de restrictions sur les rencontres admissibles. Cependant, dans tous les cas, des rencontres peuvent être interdites si des participants n'ont pas d'intérêt à se rencontrer. On peut également envisager de donner une priorité aux rencontres les plus intéressantes. Certains participants peuvent aussi arriver en retard ou indiquer des créneaux d'indisponibilité pendant lesquels ils ne peuvent pas effectuer de rencontres. On peut envisager la possibilité de créneaux de durée non unitaire si certains rendez-vous nécessitent plus de temps que d'autres. Dans certains cas, le lieu des rendez-vous est trop étendu pour que le déplacement entre deux rencontres quelconques se fasse rapidement. Il est alors possible d'ajouter des contraintes limitant les déplacements entre les rendez-vous ou allouant un temps spécifique à ces déplacements.

Selon le problème étudié, l'objectif peut être de maximiser le nombre de rencontres sur un nombre de créneaux fixés. On peut chercher à maximiser la satisfaction des participants en donnant priorité aux rencontres qu'ils demandent en premier. On peut également souhaiter minimiser les créneaux d'attente des participants.



Dans cette thèse, nous nous intéressons à un cas particulier de problème d'ordonnancement de rendez-vous en tête-tête dont l'application est l'organisation de soirées de speed-dating. Nos travaux sont menés en collaboration avec la société [lespeeddating.com](http://lespeeddating.com) qui organise des soirées de speed-dating. Nous décrivons les spécificités de ce problème dans la section suivante.

## 1.2 Caractéristiques du problème de speed-dating

Une soirée de speed-dating organisée par [lespeeddating.com](http://lespeeddating.com) permet à des hommes et des femmes de se rencontrer durant de courts créneaux. Chaque créneau horaire a une durée de 7 min. On a, dans ce cas, des contraintes spécifiques :

- certaines rencontres sont interdites si deux participants se sont déjà rencontrés lors d'une soirée précédente ;
- certains participants arrivent en retard et ne sont donc pas disponibles durant les premiers créneaux de la soirée.

La soirée se termine lorsque toutes les rencontres possibles ont eu lieu. L'objectif est de satisfaire au mieux les participants en minimisant le plus grand nombre de créneaux d'attente.

On distingue deux types de problèmes de speed-dating :

- les problèmes de type dynamique dans lesquels on ne connaît la date d'arrivée des participants qu'au fur et à mesure de la soirée ;
- les problèmes de type statique dans lesquels on considère que l'on connaît tous les paramètres de la soirée à l'avance, en particulier les dates d'arrivée de tous les participants.

Dans cette thèse, nous ne considérons que le cas statique.

## 1.3 Notations

Nous proposons dans cette section des notations pour les problèmes de speed-dating basées sur les notations classiques utilisées dans les problèmes d'ordonnancement. Nous présentons également deux représentations équivalentes d'une solution.

### 1.3.1 Données

- $\mathcal{M} := \{M_0, M_1, \dots, M_{m-1}\}$  ("Male") est l'ensemble des hommes, ( $m = |\mathcal{M}|$ ).
- $\mathcal{F} := \{F_0, F_1, \dots, F_{n-1}\}$  ("Female") est l'ensemble des femmes ( $n = |\mathcal{F}|$ ).

Pour  $i \in \llbracket 0, m-1 \rrbracket$  et  $j \in \llbracket 0, n-1 \rrbracket$ , on définit les données suivantes :

- À chaque participant  $M_i$  (resp.  $F_j$ ) est associée une date d'arrivée (ou date de disponibilité) notée  $r_i^{(1)}$  (resp.  $r_j^{(2)}$ ). Le paramètre  $r_i^{(1)}$  ou  $r_j^{(2)}$  indique le créneau à partir duquel on peut planifier des rencontres pour cette personne.
- La matrice des rencontres  $P = (p_{ij})_{(i,j) \in \mathcal{M} \times \mathcal{F}}$  représente l'ensemble des rencontres à planifier.  

$$p_{ij} := \begin{cases} 0 & \text{si la rencontre entre } M_i \text{ et } F_j \text{ est interdite,} \\ 1 & \text{sinon.} \end{cases}$$

On déduit de ces données les valeurs suivantes :

- Dans le cas idéal où l'homme  $M_i$  (resp. la femme  $F_j$ ) n'a pas d'attentes, sa date de fin au plus tôt (earliest finishing date) est  $e_i^{(1)} = r_i^{(1)} + \text{card}\{j \in \llbracket 0, n-1 \rrbracket \mid p_{ij} = 1\}$  (resp.  $e_j^{(2)} = r_j^{(2)} + \text{card}\{i \in \llbracket 0, m-1 \rrbracket \mid p_{ij} = 1\}$ ).

- Pour chaque rencontre  $(M_i, F_j) \in \mathcal{M} \times \mathcal{F}$  telle que  $p_{ij} = 1$ , on définit une date de disponibilité notée  $r_{ij}$  et une date d'échéance (due date) notée  $d_{ij}$ . La date de disponibilité de  $(M_i, F_j)$  est la plus petite date à laquelle elle peut être planifiée, c'est-à-dire  $r_{ij} = \max(r_i^{(1)}, r_j^{(2)})$ . Sa date d'échéance est la plus grande date avant laquelle elle peut être affectée sans engendrer de créneaux d'attente pour au moins l'un des deux participants (celui qui a la plus petite date d'échéance), c'est-à-dire  $d_{ij} = \min(e_i^{(1)}, e_j^{(2)})$ . On définit la date  $d_{\min}$  comme étant la plus petite date d'échéance de toutes les rencontres, c'est-à-dire :  $d_{\min} = \min\{d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\} = \min\{e_i^{(1)}, e_j^{(2)} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket\}$ .

Enfin, on note  $\mathcal{S} = \llbracket 0, s-1 \rrbracket$ , l'ensemble des créneaux de la soirée. On peut borner  $s$  naïvement par  $\max\{r_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\} + m \times n$  en affectant une rencontre par créneau à partir de la date à laquelle sont arrivés tous les participants.

### 1.3.2 Variables

Pour  $i \in \llbracket 0, m-1 \rrbracket$  et  $j \in \llbracket 0, n-1 \rrbracket$ , on définit les variables suivantes :

- La date de fin de la rencontre  $(M_i, F_j)$  est notée  $C_{ij}$  (pour  $i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1$ ).
- La date de fin des rencontres de l'homme  $M_i$  (resp. la femme  $F_j$ ) est notée  $C_i^{(1)} := \max_{j \in \llbracket 0, n-1 \rrbracket, p_{ij}=1} \{C_{ij}\}$  (resp.  $C_j^{(2)} := \max_{i \in \llbracket 0, m-1 \rrbracket, p_{ij}=1} \{C_{ij}\}$ ).
- Le nombre de créneaux d'attente de l'homme  $M_i$  (resp. la femme  $F_j$ ) est donc  $W_i^{(1)} = C_i^{(1)} - e_i^{(1)}$  (resp.  $W_j^{(2)} = C_j^{(2)} - e_j^{(2)}$ ).
- Les nombres maximaux d'attentes des hommes et des femmes se notent respectivement  $W_{\max}^{(1)} = \max\{W_i^{(1)} \mid i \in \llbracket 0, m-1 \rrbracket\}$  et  $W_{\max}^{(2)} = \max\{W_j^{(2)} \mid j \in \llbracket 0, n-1 \rrbracket\}$ .
- Notre objectif s'écrit ainsi :  $W_{\max} = \max(W_{\max}^{(1)}, W_{\max}^{(2)})$ . L'objectif  $W_{\max}$  s'apparente à un objectif de type plus grand retard  $T_{\max}$  ou  $L_{\max}$ , cependant, les dates d'échéance  $d_{ij}$  ne sont pas arbitraires car elles dépendent des dates d'arrivée de  $M_i$  et  $F_j$  et de leur nombre de rencontres et concernent les deux types de "ressources" (hommes et femmes).

### 1.3.3 Notations à trois champs $\alpha|\beta|\gamma$

Nous proposons d'étendre la notation standard en ordonnancement  $\alpha|\beta|\gamma$  (type de problème | contraintes | fonction objectif) introduite dans (Graham et al., 1979) de la façon suivante :

- Le champ  $\alpha$  vaut 1-1MS pour les problèmes d'ordonnancement de rendez-vous en tête-à-tête (one-to-one meeting scheduling).
- Le champ  $\beta$  peut prendre pour valeur :
  - $G$  le graphe de rencontres à planifier. Les sommets représentent les participants et les arêtes les rencontres devant être planifiées. S'il y a deux types de participants, le graphe  $G$  est biparti, il est noté :
    - \*  $(U \cup V, E)$  pour un graphe biparti quelconque (possibilité de rencontres interdites) ;
    - \*  $(U \cup V, E), |U| = |V|$  pour un graphe biparti avec le même nombre de participants dans chaque population ;
    - \*  $K_{m,n}$  pour un graphe biparti complet (toutes les rencontres sont obligatoires) ;
    - \*  $K_{m,m}$  pour un graphe biparti complet avec le même nombre de participants dans chaque population.
  - $r_i^{(1)}$  et/ou  $r_j^{(2)}$  si des participants peuvent arriver en retard dans le cas avec deux populations distinctes.
- Enfin,  $\gamma = W_{\max}$  (nombre maximal de créneaux d'attente des participants) est l'objectif à minimiser que nous retenons.

Par exemple, le problème  $1-1MS|K_{m,n}, r_i^{(1)}, r_j^{(2)}|W_{\max}$  fait référence au problème d’ordonnancement de rendez-vous en tête-à-tête avec deux types de populations de tailles différentes où toutes les rencontres doivent être effectuées et les participants des deux populations peuvent arriver en retard. L’objectif est de minimiser le nombre maximal de créneaux d’attente des participants.

Les problèmes de speed-dating sont donc des problèmes d’ordonnancement de rendez-vous en tête-à-tête particuliers avec deux types d’acteurs, avec dates de disponibilité pour les deux types d’acteurs, à temps d’exécution unitaire, et pour lesquels l’objectif considéré est la minimisation de l’attente maximal.

Ces notations peuvent également être étendues à d’autres problèmes que nous évoquerons dans le chapitre 2 consacré à l’état de l’art.

### 1.3.4 Représentation d’une solution

Nous décrivons ici deux représentations de solutions que nous utiliserons dans la suite de ce document. On considère l’instance suivante, à trois hommes et quatre femmes :

- liste des retards des hommes :  $r_0^{(1)} = 0$  ;  $r_1^{(1)} = 1$  et  $r_2^{(1)} = 2$  ;
- liste des retards des femmes :  $r_0^{(2)} = r_1^{(2)} = 0$  et  $r_2^{(2)} = r_3^{(2)} = 1$  ;
- liste des rencontres :  $p_{ij} = \begin{cases} 0 & \text{si } (i, j) \in \{(2, 0), (1, 1), (2, 3)\}, \\ 1 & \text{sinon.} \end{cases}$

Deux représentations d’une même solution optimale pour cette instance sont données dans les tableaux 1.1 et 1.2.

Dans le tableau 1.1, on représente en ligne les dates de rendez-vous de chaque homme et en colonne les dates de rendez-vous de chaque femme. À l’intersection de la ligne de l’homme  $M_i$  avec la colonne de la femme  $F_j$  ( $(i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket$ ), se trouve donc la date de rencontre de ces deux participants ; une rencontre interdite est indiquée par une case noire. Les dernières lignes et colonnes indiquent le nombre de créneaux d’attente du participant correspondant.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1	1	$W_i^{(1)}$
$M_0$	0	2	0	1	3	0
$M_1$	1	1		3	2	0
$M_2$	2		2	4		1
	$W_j^{(2)}$	1	1	1	1	$W_{\max} = 1$

Tableau 1.1 – Première représentation d’une solution.

Dans le tableau 1.2, on représente la même solution sous une forme proche du diagramme de Gantt bien connu en ordonnancement. Le planning de chaque participant est représenté en ligne. Une case gris clair en début de ligne indique que le (la) participant(e) n’est pas encore arrivé(e) à cette date. Une case gris foncé en fin de ligne indique que le (la) participant(e) a terminé ses rencontres et a quitté la soirée. Un tiret indique un créneau d’attente pour le (la) participant(e). Cette représentation est moins synthétique que la précédente mais permet de visualiser les attentes de chaque participant au cours de la soirée.

## 1.4 Génération d’instances

Afin de tester les méthodes de résolution que nous proposons dans les chapitres suivants, nous avons généré des instances en nous basant sur des données réelles fournies par notre partenaire. Nous décrivons ici les caractéristiques de

Participants\Créneaux	0	1	2	3	4	$W_i^{(1)}$
$M_0$	$F_1$	$F_2$	$F_0$	$F_3$		0
$M_1$		$F_0$	$F_3$	$F_2$		0
$M_2$			$F_1$	-	$F_2$	1
						$W_j^{(2)}$
$F_0$	-	$M_1$	$M_0$			1
$F_1$	$M_0$	-	$M_2$			1
$F_2$		$M_0$	-	$M_1$	$M_2$	1
$F_3$		-	$M_1$	$M_0$		1
				$W_{\max}$		1

Tableau 1.2 – Seconde représentation d’une solution.

ces instances.

La taille de l’instance est le nombre total de participants (femmes et hommes). Trois types de paramètres nous intéressent : la proportion entre le nombre de femmes et d’hommes, les retards des participants et la proportion de rencontres interdites. Nous générons des instances de taille 14, 24, 40, 50 et 70 participants. La proportion de femmes est de 40%, 50% ou 60% du nombre total de participants. Conformément aux instances réelles, on considère qu’en moyenne, 3% des rencontres sont interdites (avec un écart type de 1,4 %). Ces paramètres sont résumés dans le tableau 1.3.

	Paramètres	valeurs possibles
	nombre de participants	14, 24, 40, 50, 70
	proportion de femmes	40%, 50%, 60%
	proportion moyenne de rencontres interdites	3%

Tableau 1.3 – Plage de variation des paramètres

Le tableau 1.4 présente la proportion moyenne de chaque type de participants par date d’arrivée selon leur sexe. Nous avons généré 60 instances par taille de population (20 pour chaque proportion d’hommes/femmes).

Date d’arrivée	Proportion	
	Hommes	Femmes
0	55%	70%
1	20%	14%
2	12%	6%
3	6%	4%
4	4%	3%
5	2%	2%
6	1%	1%

Tableau 1.4 – Proportion moyenne de participants par date d’arrivée

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté et défini les problèmes d’ordonnancement de rendez-vous en tête-à-tête et proposé plusieurs cas d’applications. En particulier, nous avons explicité le problème que nous allons étudier et pour lequel nous allons développer des méthodes de résolution. Nous avons proposé des notations pour ces problèmes. Dans le chapitre suivant, nous nous intéressons aux différentes applications de ces problèmes traitées dans la littérature et recensons également différents problèmes classiques se rapprochant du problème qui nous intéresse.



## Chapitre 2

# État de l'art

### Sommaire

<b>2.1</b>	<b>Introduction</b>	<b>21</b>
<b>2.2</b>	<b>Problèmes d'ordonnancement de rendez-vous en tête-à-tête</b>	<b>21</b>
<b>2.3</b>	<b>Autres problèmes de rencontres</b>	<b>23</b>
<b>2.4</b>	<b>Problèmes proches du problème de speed-dating</b>	<b>24</b>
<b>2.5</b>	<b>Conclusion</b>	<b>27</b>

### 2.1 Introduction

Dans ce chapitre, nous présentons des problèmes étudiés dans la littérature entrant dans la catégorie des problèmes d'ordonnancement de rendez-vous en tête-à-tête tels que définis dans le chapitre précédent. Nous donnons également un aperçu de problèmes classiques d'ordonnancement ayant des similitudes avec le problème de speed-dating.

### 2.2 Problèmes d'ordonnancement de rendez-vous en tête-à-tête

La littérature compte plusieurs cas d'étude de problèmes d'ordonnancement en tête-à-tête même si ceux-ci ne sont pas classifiés. Nous référençons des articles qui étudient différents cas de ce type de problèmes et montrons comment ils peuvent entrer dans notre classification. Nous présentons leur complexité lorsqu'elle est connue et les approches qui ont été employées pour les résoudre.

Dans (Rinaldi et Serafini, 2007), les auteurs étudient un problème d'ordonnancement de rendez-vous en tête-à-tête entre environ 60 parents et 30 professeurs. Une liste de rencontres est fixée à l'avance, celle-ci est élaborée à partir du choix de rencontres des parents. Les objectifs sont de minimiser la durée de l'événement et les temps d'attente des parents, ceux-ci sont traités de manière lexicographique. Si l'on reprend les notations que nous avons proposées, ce problème s'écrit  $1-1MS|(U \cup V, E)|C_{\max}, \gamma$  où  $\gamma$  est un objectif de minimisation des temps d'attente des parents uniquement. À noter que cet objectif supplémentaire n'est pas un objectif  $L_{\max}$  car les parents n'ont pas de date d'échéance prédéterminée et que leur date d'arrivée n'est prise en compte qu'à travers la date fixée pour le premier rendez-vous. Les créneaux considérés comme attente sont les créneaux sans rencontre entre le premier rendez-vous et le dernier. Les auteurs proposent une extension de ce problème dans lequel certaines rencontres peuvent durer deux fois le temps de base. Ce problème d'ordonnancement de rencontres parents-professeurs est  $\mathcal{NP}$ -difficile quelle que soit la durée des rencontres. Ce résultat provient d'une réduction du problème  $\mathcal{NP}$ -difficile d'open shop avec contraintes de non-attentes :  $O|no-wait, p_{ij} \in \{0, 1\}|C_{\max}$ . Les auteurs résolvent ce problème de manière approchée en deux phases. Dans un premier temps, ils résolvent le problème de minimisation de fin de soirée de manière exacte par recherches successives de couplages maximums dans un graphe biparti. Dans un second temps, ils minimisent les attentes des parents par une recherche locale en échangeant deux rencontres d'un parent ou deux rencontres d'un professeur.

Dans (Ernst et al., 2003), les auteurs implémentent un outil d'organisation de rencontres en tête-à-tête lors de salons commerciaux pour la promotion du tourisme australien. Environ 600 vendeurs (groupes hôteliers, voyagistes...) et 700 acheteurs (agences de voyages...) se rencontrent. Chaque participant choisit une liste de participants de l'autre type triée par ordre de préférence. Les organisateurs en déduisent un indice de satisfaction pour chaque rencontre. L'objectif est de maximiser le nombre de rencontres ainsi que la satisfaction totale des participants. Chaque vendeur est affecté à un lieu fixé et étant donnée la taille de l'événement, il n'est pas possible pour un acheteur de se déplacer d'un vendeur à n'importe quel autre en trois minutes (temps imparti entre deux créneaux de rencontres). Les auteurs ont donc ajouté des contraintes de déplacements qui obligent deux rencontres successives à être positionnées suffisamment proches l'une de l'autre. Ce problème est un cas particulier de  $1-1MS|(U \cup V, E)|_{\gamma_1, \gamma_2}$  avec des contraintes limitant les déplacements et où les objectifs  $\gamma_1$  et  $\gamma_2$  sont respectivement la maximisation du nombre de rencontres et la maximisation de la somme des indices de satisfaction des rencontres planifiées. Ce problème, sans les contraintes de déplacement, est similaire à un sous-problème d'affectation en trois dimensions (three-dimensional assignment problem) polynomial car la satisfaction engendrée par une rencontre ne dépend pas de l'heure à laquelle est affectée la rencontre. Les auteurs résolvent ce problème séquentiellement : pour chaque créneau de rencontres, sont déterminées les rencontres possibles (qui n'ont pas déjà été affectées et réalisables en terme de déplacements) puis le problème d'affectation (problème de couplage maximum de poids maximal dans un graphe biparti) induit par ces rencontres est résolu.

Dans (Huang et al., 2012), les auteurs développent un programme de planification de "business dating". Ce problème est le même que le problème précédent sans les contraintes de déplacement. Des vendeurs et des acheteurs se rencontrent. Le nombre de participants est variable, des applications réelles testées par les auteurs contiennent entre 30 et 42 vendeurs et de 46 à 84 acheteurs. De manière préliminaire, chacun des participants fournit à l'organisateur une liste des rencontres qu'il souhaite effectuer par ordre de priorité dont sont déduits des indices de préférence pour chaque rencontre. L'objectif est également de maximiser le nombre de rencontres et de maximiser la satisfaction totale des participants. Ce problème se note  $1-1MS|(U \cup V, E)|_{\gamma_1, \gamma_2}$  avec  $\gamma_1$  et  $\gamma_2$  définis de la même manière que précédemment. Bien qu'un algorithme polynomial soit connu pour ce problème, les auteurs utilisent une approche heuristique plus facile à implémenter. De la même manière que dans (Ernst et al., 2003), ils résolvent successivement des problèmes d'affectation pour chaque créneau de rencontres.

Dans (Gandibleux et al., 2006) and (Guéret et al., 2009), les auteurs s'intéressent à un problème d'ordonnement de rencontres dans le cadre d'une bourse d'échanges de technologies. Dans ce type d'événement, entre 100 et 200 participants (académiques et industriels) se rencontrent. Chaque participant fournit une liste de priorités sur les rencontres qu'il souhaite effectuer. De nombreuses contraintes additionnelles sont prises en compte. Notamment, il y a un nombre limite de rencontres pouvant avoir lieu en même temps et chaque participant a un nombre maximal de rencontres consécutives qu'il peut effectuer. Plusieurs objectifs sont considérés : maximiser le nombre de rencontres planifiées, maximiser les premiers choix des participants, minimiser le coût de location des salles et minimiser la distance totale de trajet entre les rencontres entre autres. Ce problème est un cas particulier de  $1-1MS|(U \cup V, E)|_{\gamma}$  avec de nombreuses contraintes additionnelles et où  $\gamma$  agrège plusieurs objectifs. Les auteurs de (Gandibleux et al., 2006) montrent que ce problème est une généralisation du problème  $\mathcal{NP}$ -difficile de set-packing. Ils modélisent ce problème à l'aide d'un programme linéaire en nombres entiers. Les auteurs de (Guéret et al., 2009) proposent de résoudre ce problème en trois phases : maximiser le nombre de rencontres planifiées (par PLNE), minimiser le coût des salles louées (algorithme de programmation dynamique) et minimiser les déplacements entre les tables (recherche taboue).

Dans (Bartholdi et McCroan, 1990), les auteurs développent une application pour planifier des entretiens d'embauche entre des étudiants et des entreprises dans le domaine du droit dans un forum pour l'emploi. Entre 25 et 50 entreprises et 100 à 200 étudiants participent au "Southern Public Interest Job Fair" chaque année, événement se tenant pendant deux jours. Les étudiants et les recruteurs déterminent à l'avance les rencontres qu'il souhaitent effectuer. Le nombre de créneaux disponibles pour les entretiens étant limité à 23, chacun des participants peut former une liste de 23 rencontres souhaitées au maximum. L'application ordonnance l'ensemble des rencontres (demandées par les deux parties) avec pour objectif de créer des plannings compacts à la fois pour les étudiants et les recruteurs. Le problème de base se modélise comme un problème de coloration d'arêtes dans un graphe biparti. Il est toujours possible de planifier toutes les rencontres demandées d'après le théorème de König (König, 1916). Ce problème est

polynomial sans recherche de compacité des plannings. Les auteurs résolvent ce problème par l’affectation séquentielle des rencontres ; si dans ce processus, une infaisabilité est détectée, la solution est réparée par une recherche de chemin augmentant dans le graphe biparti. Ce problème se note  $1-1MS|(U \cup V, E)|\gamma$  où  $\gamma$  est un objectif de compacité des plannings des participants (cependant, cet objectif n’est pas explicité dans la modélisation proposée).

Dans (Bofill et al., 2014), les auteurs s’intéressent à l’organisation de rendez-vous d’affaire tels que des forums scientifiques et technologiques faisant se rencontrer des chercheurs, des entreprises et des investisseurs dans le but de développer des partenariats industriels et/ou scientifiques. Chacun des participants indique à l’organisateur la liste des autres participants qu’il souhaite rencontrer. Le problème consiste à planifier toutes les rencontres sur un nombre de créneaux fixé et de positionner chacune de ses rencontres à une table. Le premier objectif est de minimiser le nombre total de créneaux d’attente des participants (les créneaux d’attente sont déterminés entre le premier et le dernier rendez-vous d’un participant). Un deuxième objectif traité de manière lexicographique est considéré, il s’agit de minimiser le nombre total de déplacements des participants entre les tables. Ce problème est de type  $1-1MS||\gamma, \gamma'$  où  $\gamma$  est l’objectif de minimisation des attentes (défini de la même manière que dans le problème de rencontres parents-professeurs) et  $\gamma'$  est l’objectif de minimisation de la somme des déplacements des participants. Ce problème est  $\mathcal{NP}$ -difficile par réduction du problème d’emploi du temps (RTT) (Even et al., 1976). Les auteurs modélisent ce problème à l’aide de deux modèles de programmation par contraintes, le deuxième n’utilisant que des variables booléennes. Ces modèles sont testés sur des instances réelles dont le nombre de participants varie de 42 à 78 pour un nombre de rencontres compris entre 125 et 302. Des comparaisons sont faites entre les résultats obtenus par différents solveurs de programmation par contraintes et des solutions construites “à la main”.

Dans (Katz et al., 2011), les auteurs étudient un problème de type speed-dating. Dans ces travaux, chacun des participants remplit un formulaire sur le genre de personnes qu’elle/il recherche. L’organisateur en déduit un indice de priorité pour chaque rencontre. Afin d’obtenir un planning équitable pour tous les participants, une borne inférieure et une borne supérieure sont fixées sur le nombre de rencontres que peut effectuer un participant. Le premier objectif est de minimiser le nombre de violations de cette contrainte, le deuxième objectif est de maximiser la satisfaction des participants en favorisant les rencontres prioritaires. Dans ce travail, les auteurs considèrent à la fois le speed-dating hétérosexuel (avec deux populations) et homosexuel (avec une seule population). Ces deux problèmes sont respectivement des cas particuliers de  $1-1MS|(U \cup V, E)|\gamma_1, \gamma_2$  et  $1-1MS||\gamma_1, \gamma_2$  avec  $\gamma_1$  et  $\gamma_2$  définis comme précédemment et contraintes supplémentaires sur le nombre minimal et maximal de rencontres à effectuer par participants. Le problème avec une population est  $\mathcal{NP}$ -complet par réduction du problème de coloration des arêtes dans un graphe, cependant le cas biparti est polynomial. Les auteurs résolvent le cas biparti en trois étapes. Dans un premier temps, une recherche dichotomique est effectuée sur la valeur maximale de la différence entre le nombre minimal de rencontres requises et le nombre de rencontres effectuées par participants. Ensuite, les rencontres à effectuer sont déterminées à l’aide d’un problème de flot à coût maximal. Enfin, les dates des rencontres sont calculées par la résolution de coloration d’arêtes dans un graphe biparti. Les auteurs déduisent de cette méthode un algorithme d’approximation pour le cas général du problème. Les auteurs proposent également un algorithme glouton de résolution du problème général. À chaque étape de l’algorithme une rencontre est choisie et une date lui est affectée si c’est possible. Les rencontres qui concernent un participant qui ne respecte pas la contrainte du nombre de rencontres minimal sont choisies en priorité.

Nous avons ainsi inventorié différents cas de problèmes d’ordonnancement de rendez-vous en tête-à-tête. Ceux-ci ont en commun l’organisation de rencontres entre personnes lors d’une session unique. Dans la majorité des cas, deux populations distinctes se rencontrent et la durée des créneaux est unitaire. Cependant, ils présentent des caractéristiques différentes dues aux spécificités métiers (priorités sur les rencontres, limitation des déplacements entre deux créneaux, durées non unitaires des rencontres).

## 2.3 Autres problèmes de rencontres

Nous référençons ici quelques problèmes que nous ne considérons pas comme des problèmes d’ordonnancement de rendez-vous en tête-à-tête bien qu’ils possèdent certains points en commun avec ceux-ci. Nous précisons les raisons pour lesquelles ils n’entrent pas dans la classification.

Évoquons, tout d’abord, le problème de “social golfer” (posté sur `sci.op-research` en mai 1998, voir



aussi (Colbourn et Nonay, 1997)) ainsi défini : “Le but est de planifier  $g \times p$  golfeurs en  $g$  groupes de  $p$  joueurs pendant  $w$  semaines de sorte que deux golfeurs ne jouent pas plus d’une fois dans un groupe en commun”. Dans ce problème, le nombre de participants par groupe est en général supérieur à deux, ce problème ne peut donc pas être considéré comme un problème d’ordonnancement de rendez-vous en tête-à-tête.

Les problèmes d’organisation de tournoi tels que ceux définis dans (de Werra, 1988) ou (Easton et al., 2001) par exemple ne peuvent pas non plus être considérés comme des problèmes d’ordonnancement de rendez-vous en tête-à-tête : dans ce type de problème, les rencontres sont organisées dans des lieux différents (matchs à domicile ou à l’extérieur) et l’objectif est en général la minimisation des distances de trajet entre les rencontres.

D’autres problèmes appelés problèmes d’ordonnancement de rendez-vous (meeting scheduling problem) comme dans (Sen et Durfee, 1998) sont également différents des problèmes d’ordonnancement de rendez-vous en tête-à-tête que nous traitons. Dans ce type de problème, le nombre de participants présents lors d’une rencontre est en général supérieur à deux. Par ailleurs, les rendez-vous sont dispersés dans le temps, il n’y a pas de session unique de rendez-vous et deux rendez-vous ont rarement lieu en même temps. L’objectif principal de ces problèmes est de planifier quelques rendez-vous de durée variable concernant plus de deux participants sur un horizon de temps assez large dans le but de maximiser la satisfaction des participants.

## 2.4 Problèmes proches du problème de speed-dating

Malgré les problèmes voisins évoqués dans ce chapitre, nous avons vu que le problème de speed-dating tel que nous le définissons est récent et a peu été traité dans la littérature. À notre connaissance, les seuls travaux sur le sujet sont ceux de (Lapègue et al., 2011). Dans ces travaux, les auteurs proposent un modèle de programmation par contraintes pour résoudre le cas dynamique du problème de speed-dating avec contraintes additionnelles de déplacement. Cependant, notre problème se rapproche de différents problèmes classiques de la littérature que nous listons dans cette section.

### 2.4.1 Problème d’open shop

Dans le problème d’open-shop, on doit réaliser  $n$  jobs  $J_1, \dots, J_n$ . Pour cela, on dispose de  $m$  machines  $M_1, \dots, M_m$ . Chaque job  $J_i$  est composé de  $m$  opérations  $O_{i1}, \dots, O_{im}$ . L’opération  $O_{ij}$  ( $i \in \llbracket 1, n \rrbracket$  et  $j \in \llbracket 1, m \rrbracket$ ) est effectuée sur la machine  $M_j$  et a une durée d’exécution  $p_{ij}$ . L’ordre des opérations d’un job est libre. Une machine ne peut effectuer qu’une opération à la fois et plusieurs opérations d’un même job ne peuvent être réalisées en même temps. On s’intéresse au cas particulier où les opérations ont une durée unitaire et dans lequel un job  $J_i$  peut avoir une date de disponibilité (ou date d’arrivée)  $r_i$  (release date) avant laquelle il ne peut être effectué et une date d’échéance  $d_i$  (due date) qui est sa date de fin demandée. Une fois le problème résolu, chaque job  $J_i$  a une date de fin  $C_i$ , et on en déduit le retard algébrique du job  $J_i$  noté  $L_i = C_i - d_i$ . Ce retard est négatif si le job est terminé avant sa date d’échéance. L’objectif qui nous intéresse ici est la minimisation du plus grand retard algébrique :  $L_{\max} = \max_{i \in \llbracket 1, n \rrbracket} \{L_i\}$ .

Le problème de speed-dating  $1\text{-}1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  (cas où les femmes sont à l’heure et toutes les rencontres sont obligatoires) s’apparente au problème d’open shop  $O|r_i, p_{ij} = 1|L_{\max}$  dans lequel chaque machine est associée à une femme, chaque job à un homme et chaque opération représente une rencontre. Chaque job a une date de disponibilité égale à la date d’arrivée de l’homme associé. Le problème d’open shop  $O|r_i, p_{ij} = 1|L_{\max}$  est de complexité  $O(n^{5/2}m^{3/2} \log(mn))$  (Kellerer et al., 1995).

Cependant, dans le cas général où les hommes et les femmes arrivent en retard, les objectifs  $W_{\max}$  et  $L_{\max}$  diffèrent car dans le cas de l’open shop, les machines n’ont pas de date de disponibilité. Si les créneaux d’attente des hommes peuvent être assimilés aux retards des jobs, les créneaux d’attente des femmes n’ont pas d’équivalent dans les problèmes d’open shop. De plus, les dates d’échéance des participants ne sont pas quelconques (elles dépendent à la fois de la date d’arrivée du participant et de son nombre de rencontres).

### 2.4.2 Problème d'emploi du temps

La version basique du problème d'emploi du temps peut s'énoncer de la manière suivante : soient les ensembles  $\mathcal{C}$ ,  $\mathcal{P}$  et  $\mathcal{H}$  représentant respectivement l'ensemble des classes, l'ensemble des professeurs et l'ensemble des créneaux horaires considérés. Toute classe doit suivre son propre programme qui se compose d'un ensemble d'enseignements. Chaque enseignement est assuré par un unique professeur pendant un créneau horaire disponible. On définit la matrice  $R \in \mathcal{M}_{|\mathcal{P}| \times |\mathcal{C}|}(\mathbb{N})$  ; pour  $(i, j) \in \mathcal{P} \times \mathcal{C}$ ,  $R_{ij}$  est le nombre d'heures d'enseignement que le professeur  $i$  doit effectuer avec la classe  $j$ . Il s'agit de construire, si cela est possible, un emploi du temps sur un horizon donné et commun aux classes et aux professeurs, tel que :

- chaque professeur n'assure qu'un seul enseignement à la fois ;
- chaque classe n'assiste qu'à un seul enseignement à la fois ;
- chaque classe effectue tout son programme au cours de l'horizon donné.

À cela, s'ajoutent des contraintes d'indisponibilité de professeurs ou de classes pour certains créneaux horaires. Nous notons ce problème de construction d'emploi du temps  $TT$  (Time Tabling). Le problème  $TT$  est  $\mathcal{NP}$ -complet même si  $R_{ij} \in \{0, 1\}$  (Even et al., 1976).

Le lien entre  $TT$  et les problèmes de speed-dating peut s'établir de la manière suivante : les hommes correspondent aux classes, les femmes aux professeurs et les rencontres aux enseignements. Ainsi le programme d'une classe correspond aux rencontres d'un homme. De plus, on est dans le cas particulier où  $\forall (i, j) \in \mathcal{P} \times \mathcal{C}$ ,  $R_{ij} \in \{0, 1\}$  car chaque couple de participants ne se rencontre au plus qu'une fois. On considère les retards des participants comme des créneaux de non disponibilité. Notre problème est un cas particulier de  $TT$  car les indisponibilités de chaque participant (liées à son retard) sont regroupées au début du planning.

La principale différence entre les problèmes de speed-dating et d'emploi du temps réside dans le choix de la fonction objectif. Dans les problèmes de speed-dating, on cherche à minimiser le nombre d'attentes alors que dans le problème  $TT$ , on cherche à minimiser le nombre de créneaux horaires.

### 2.4.3 Problèmes d'ordonnancement par fournées avec incompatibilités entre les jobs

On décrit le problème d'ordonnancement par fournées avec incompatibilités entre les jobs (*batch scheduling with job incompatibilities*) de la manière suivante : on doit planifier l'exécution de  $n$  jobs  $J_1, \dots, J_n$  par fournées sur une unique machine qui peut effectuer plusieurs jobs simultanément. Une fournée est composée d'un sous-ensemble de  $\{J_1, \dots, J_n\}$ . Chaque job  $J_i$  a un temps d'exécution  $p_i$ , une date de disponibilité  $r_i$  et une date d'échéance  $d_i$ . Il existe des incompatibilités entre les jobs, c'est à dire que certains ne peuvent pas être exécutés en même temps. On modélise ces contraintes par un graphe de compatibilité  $G = (V, E)$  dans lequel les jobs sont représentés par les sommets et une arête représente deux jobs pouvant être exécutés en même temps. La machine peut avoir une capacité finie notée  $b$  (nombre maximum de jobs pouvant être accueillis simultanément sur la machine) ou infinie si  $b \geq n$ .

Le problème de speed-dating sans retards pour les participants peut être modélisé comme un problème d'ordonnancement par fournées sur une machine à capacité infinie avec incompatibilités entre jobs. Les jobs représentent les rencontres réalisables. Le graphe de compatibilité est défini de la manière suivante :  $G = (V, E)$  où  $V = \{v_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$  est l'ensemble des rencontres réalisables et  $E = \{(v_{i_1, j_1}, v_{i_2, j_2}) \in V^2 \mid i_1 \neq i_2 \text{ et } j_1 \neq j_2\}$  (deux rencontres sont compatibles si les deux femmes et les deux hommes concernés sont différents). De plus, on se trouve dans le cas particulier où le temps d'exécution de chaque job est unitaire.

Dans la littérature, la principale fonction objectif étudiée pour les problèmes d'ordonnancement par fournée est la date de fin du dernier job  $C_{\max}$ . Ainsi la complexité de  $1|p\text{-batch}, G = (V, E), b \geq n, p_i = 1|C_{\max}$  est discutée dans (Boudhar et Finke, 2000) selon la structure du graphe de compatibilité des jobs. À notre connaissance, les seuls travaux concernant l'objectif  $L_{\max}$  considèrent des familles de jobs compatibles. Le graphe de compatibilité est donc une réunion de cliques ce qui n'est pas le cas des graphes obtenus pour les rencontres de speed-dating.

### 2.4.4 Coloration d'arêtes d'un graphe par intervalles

La coloration des arêtes d'un graphe par intervalles (*interval coloring* ou *consecutive coloring*) consiste à colorer les arêtes d'un graphe avec des entiers naturels de manière à ce que l'ensemble des couleurs des arêtes incidentes à un même sommet soient consécutives. Ainsi les couleurs des arêtes sortant d'un sommet forment un intervalle de  $\mathbb{N}$ . On parle de  $t$ -coloration par intervalles (*interval  $t$ -coloring*) si au moins un arc a la couleur  $i$  pour  $i \in \llbracket 1, t \rrbracket$ . Déterminer si un graphe biparti possède une  $t$ -coloration par intervalles est  $\mathcal{NP}$ -complet (Sevastjanov, 1990). Il existe certains graphes bipartis pour lesquels la détermination de la  $t$ -coloration par intervalles peut s'effectuer en temps polynomial (Asratian et Casselgren, 2007).

Considérons les problèmes de décision associés aux problèmes de speed-dating définis comme suit : “pour  $k \in \mathbb{N}$  fixé, existe-t-il un ordonnancement des rendez-vous tel que  $W_{\max} \leq k$  ?” On peut modéliser ces problèmes de décision comme des problèmes de  $t$ -coloration par intervalles. À partir d'une instance de 1-1MS|( $U \cup V, E$ )| $W_{\max}$ , on construit le graphe biparti  $G = (U' \cup V', E)$  avec  $U' = \mathcal{M} \cup \llbracket 1, kn \rrbracket$  et  $V' = \mathcal{F} \cup \llbracket 1, km \rrbracket$  où  $k$  désigne le nombre maximal d'attentes autorisées par personne ( $W_{\max} \leq k$ ). Une arête existe entre deux sommets  $M_i \in \mathcal{M}$  et  $F_j \in \mathcal{F}$  si et seulement si  $p_{ij} = 1$  et pour tout  $i \in \llbracket 0, m-1 \rrbracket$  (resp.  $j \in \llbracket 0, n-1 \rrbracket$ ), il existe une arête entre  $M_i$  (resp.  $F_j$ ) et chacun des sommets de  $\llbracket ik+1, (i+1)k \rrbracket$  (resp.  $\llbracket jk+1, (j+1)k \rrbracket$ ). Ces dernières arêtes représentent les attentes possibles des participants. On note le degré maximum du graphe  $G$  :  $\Delta_k := \max(\max_{i \in \llbracket 0, m-1 \rrbracket} \{\#\{j \in \llbracket 0, n-1 \rrbracket \mid p_{ij} = 1\}\}, \max_{j \in \llbracket 0, n-1 \rrbracket} \{\#\{i \in \llbracket 0, m-1 \rrbracket \mid p_{ij} = 1\}\}) + k$ .

Cependant, il ne suffit pas de résoudre le problème de  $\Delta_k$ -coloration de ce graphe pour obtenir une solution au problème de speed-dating car l'intervalle associé à chaque sommet (participant) ne contient pas nécessairement zéro, le participant n'est donc pas assuré de démarrer ses rencontres dès le début de la soirée. Ainsi, le problème de speed-dating que nous étudions est un cas particulier de  $\Delta_k$ -coloration.

### 2.4.5 Problème de multiflot entier

Le problème de multiflot entier dans un graphe orienté à capacité se définit de la manière suivante : on dispose du graphe orienté  $G = (V, E)$  pour lequel chaque arc  $(u, v) \in E$  a une capacité  $c(u, v) \geq 0$ . On choisit  $n$  sommets  $s_1, \dots, s_n$  sources et  $n$  sommets  $t_1, \dots, t_n$  puits dans  $V$  non nécessairement distincts. Chaque sommet puits  $t_i$  a une demande  $R_i \in \mathbb{N}$  en produit  $i$  égale à la quantité de produit  $i$  fournie par le sommet source  $s_i$  correspondant. On définit les fonctions de flot  $(f_1, \dots, f_n)$  qui vérifient les contraintes suivantes :

- $\forall (u, v) \in E, \forall i \in \llbracket 1, n \rrbracket, f_i(u, v) \in \mathbb{N}$
- $\forall (u, v) \in E, \sum_{i=1}^n f_i(u, v) \leq c(u, v)$  : le flot total traversant chaque arc est limité par la capacité de celui-ci ;
- $\forall i \in \llbracket 1, n \rrbracket, \forall u \in V \setminus \{s_i, t_i\}, \sum_{(u,v) \in E} f_i(u, v) = \sum_{(w,u) \in E} f_i(w, u)$  : loi de conservation du flot de chaque produit au passage de chaque sommet ;
- $\forall i \in \llbracket 1, n \rrbracket, \sum_{(s_i,v) \in E} f_i(s_i, v) - \sum_{(u,s_i) \in E} f_i(u, s_i) = R_i$  : la quantité de chaque produit circulant dans le réseau est égale à la demande de ce produit.

Dans un graphe orienté, le problème de multiflot entier est  $\mathcal{NP}$ -complet à partir de deux produits et pour des arcs de capacité unitaire (Even et al., 1976).

Notre problème de décision se modélise comme un problème de multiflot maximum. On construit le graphe orienté  $G = (V, E)$  de la manière suivante :

- $V = \mathcal{M} \cup \mathcal{S} \cup \mathcal{F}$  où  $\mathcal{M}$ ,  $\mathcal{S}$  et  $\mathcal{F}$  désignent respectivement l'ensemble des femmes, des créneaux et des hommes et on suppose que  $W_{\max} \leq k$  pour un entier  $k$  fixé.

- Un arc  $(M_i, t) \in \mathcal{M} \times \mathcal{S}$  existe si et seulement si l'homme  $M_i$  est disponible à la date  $t$  et un arc  $(t, F_j) \in \mathcal{S} \times \mathcal{F}$  existe si et seulement si la femme  $F_j$  est disponible à la date  $t$ . (Le nombre de créneaux accordés à chaque participant correspond au nombre de ses rencontres plus le nombre de créneaux d'attente possibles  $k$ . Les participants peuvent commencer leurs rencontres dès leur date d'arrivée.
- Il y a  $m$  produits qui représentent chacun un homme. Chaque sommet  $M_i \in \mathcal{M}$  est sommet source du produit  $s_i$  (où  $i \in \llbracket 0, m-1 \rrbracket$ ). La quantité disponible de chaque produit  $s_i$  correspond au nombre de rencontres de l'homme  $M_i$ .
- Pour  $j \in \llbracket 0, n-1 \rrbracket$ , le sommet  $F_j$  de  $\mathcal{F}$  a une demande de 1 en  $s_i$  ( $i \in \llbracket 0, m-1 \rrbracket$ ) si la femme  $F_j$  doit rencontrer l'homme  $M_i$ , sa demande est nulle sinon).
- Chaque arc a une capacité unitaire.

### Exemple 2.4.1.

Nous illustrons la construction du graphe de multiflot sur l'instance de la figure 2.1. Le graphe associé pour le problème de décision  $W_{\max} \leq 2$  est présenté dans la figure 2.2. Les triplets associés aux participants correspondent aux quantités disponibles de chaque "produit" pour les hommes et aux quantités de produit requises par les femmes.

		$F_0$	$F_1$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	1
$M_0$	0		
$M_1$	1		
$M_2$	1		

FIGURE 2.1 – Instance de speed-dating

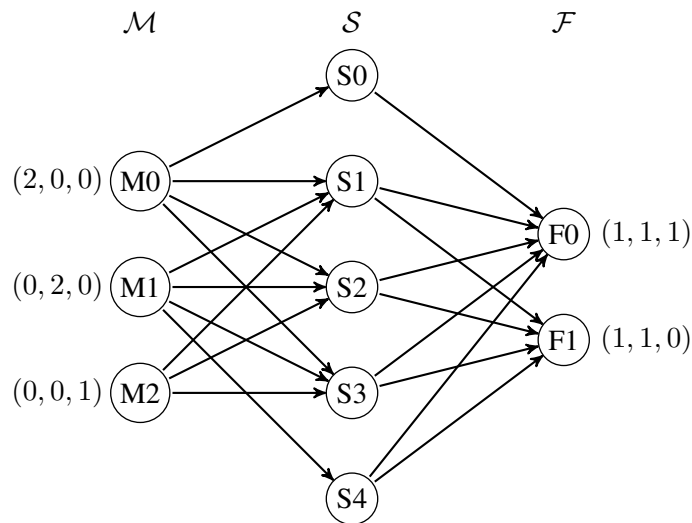


FIGURE 2.2 – Graphe de multiflot associé à l'instance de speed-dating de la figure 2.1

Le type de graphe que nous obtenons par cette modélisation est un graphe triparti particulier car le degré d'un sommet représentant un participant dépend à la fois du nombre de rencontres et du nombre de créneaux d'attente autorisés. Il s'agit donc d'un cas particulier d'un problème  $\mathcal{NP}$ -difficile.

## 2.5 Conclusion

Dans ce chapitre, nous avons passé en revue à la fois des problèmes d'ordonnancement de rendez-vous en tête-à-tête qui entrent dans notre classification ainsi que des problèmes classiques d'ordonnancement ou de graphes qui sont proches de notre problème. La principale différence entre les problèmes d'ordonnancement de rendez-vous en tête-à-tête rencontrés dans la littérature et le nôtre est la prise en compte de préférences parmi les rencontres. Dans notre

problème, il n'y a pas de nombre limite de créneaux, ainsi toutes les rencontres admises peuvent être planifiées et il n'est pas nécessaire de donner une priorité aux rendez-vous. Un point commun entre les problèmes de la littérature et le problème de speed-dating est le souhait des participants de limiter le nombre de créneaux d'attente, dans notre cas, il s'agit de la fonction objectif. En raison de ces spécificités, les méthodes de résolution de la littérature ne peuvent être appliquées directement à notre problème.

Les problèmes d'ordonnancement et de graphes présentés ici présentent également des similitudes avec le problème de speed-dating. Toutefois, aucun ne correspond exactement à notre problème. Les résultats et méthodes de résolution de ces problèmes ne sont donc pas directement applicables au nôtre.

Dans la partie suivante, nous démontrons la complexité du problème général de speed-dating ainsi que celle de plusieurs sous-cas.

**Deuxième partie**

**Résultats de complexité**



## Chapitre 3

# Résultats généraux de complexité

### Sommaire

<b>3.1 Introduction</b> . . . . .	<b>31</b>
<b>3.2 Réductions évidentes</b> . . . . .	<b>32</b>
<b>3.3 Conclusion</b> . . . . .	<b>33</b>

### 3.1 Introduction

La connaissance de la complexité théorique d'un problème permet de définir le champ d'action par rapport aux méthodes de résolution qu'il est possible d'envisager. Définir la frontière de complexité d'un problème en fonction de ses contraintes offre une meilleure connaissance de ce qui rend le problème facile ou difficile à traiter et fournit des éléments permettant de déterminer la manière de traiter chaque version du problème.

Comme nous l'avons vu dans le chapitre 2, la complexité de notre problème ne se déduit pas directement de problèmes de la littérature. Nous allons donc consacrer cette partie à la recherche de la complexité des différents sous-cas du speed-dating. Nous y démontrons différents résultats de complexité de problèmes statiques de speed-dating, c'est-à-dire des problèmes pour lesquels la date d'arrivée de chaque participant est connue à l'avance. Comme nous le verrons, les résultats diffèrent selon les contraintes considérées. Nous listons ici l'ensemble des cas envisagés.

Nous avons résumé dans le tableau 3.1 tous les cas statiques de problèmes de speed-dating. Dans la partie supérieure de ce tableau nous recensons les différents cas dans lesquels toutes les rencontres sont obligatoires. En ligne, on représente respectivement les cas avec autant d'hommes que de femmes et un nombre d'hommes et de femmes quelconques. En colonne, apparaissent respectivement les cas sans retards, avec retards autorisés pour les hommes, avec retards autorisés pour les femmes et avec retards autorisés pour tous les participants. De la même manière, nous retrouvons l'ensemble des cas dans lesquels des rencontres sont interdites dans la partie inférieure du tableau. Nous nous efforçons par la suite de déterminer la complexité de chacun d'entre eux.

	$\emptyset$	$r_i^{(1)}$	$r_j^{(2)}$	$r_i^{(1)}, r_j^{(2)}$
$K_{m,m}$				
$K_{m,n}$				
$(U \cup V, E),  U = V $				
$(U \cup V, E)$				

Tableau 3.1 – Sous-cas du problème de speed-dating pour l'objectif  $W_{\max}$ .



## 3.2 Réductions évidentes

Dans cette section, nous listons toutes les réductions immédiates entre les différents problèmes de speed-dating. Nous utiliserons ces réductions pour démontrer les corollaires des chapitres suivants.

### 3.2.1 Réductions dans le cas général

La figure 3.1 représente des réductions selon les proportions de chaque population, le retard des populations et l'interdiction des rencontres. Une réduction  $A \propto B$  s'écrit dans ce graphe :  $A \rightarrow B$ . Ce sont des réductions classiques en ordonnancement.

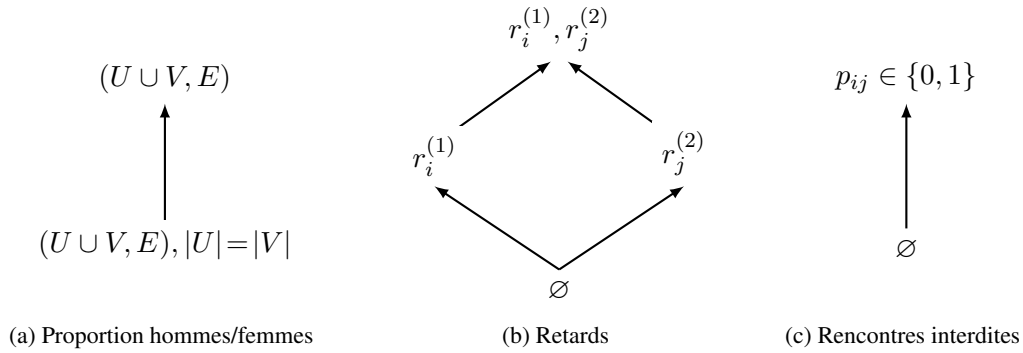


FIGURE 3.1 – Graphes de réductions

### 3.2.2 Réductions dans le cas où des rencontres sont interdites

Si des rencontres sont interdites, le cas  $(U \cup V, E)$  se réduit polynomialement à  $(U \cup V, E), |U| = |V|$  et vice versa. En effet, on peut passer du premier au deuxième en ajoutant des participants fictifs qui n'effectuent aucune rencontre quelles que soient les dates d'arrivée des participants. L'autre sens est trivial.

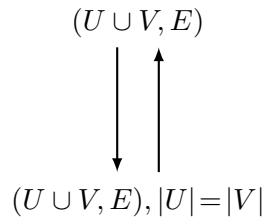


FIGURE 3.2 – Graphes de réductions avec rencontres interdites

### 3.2.3 Rôles symétriques des femmes et des hommes

Dans le cas où une seule population admet des retards, on remarque que les rôles des deux populations sont symétriques. Ainsi, par exemple, on peut réduire  $1-1MS|r_i^{(1)}|W_{\max}$  à  $1-1MS|r_j^{(2)}|W_{\max}$  (et réciproquement) en échangeant le rôle des femmes et des hommes.

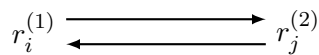


FIGURE 3.3 – Graphes de réductions avec rencontres interdites pour une population

### 3.3 Conclusion

Ce chapitre d'introduction à la complexité des problèmes de speed-dating a permis de formaliser les réductions évidentes entre différents cas du problème de speed-dating. Dans le chapitre 4, nous démontrons que plusieurs cas de notre problème sans rencontres interdites sont polynomiaux et dans le chapitre 5, nous démontrons que l'ensemble des problèmes avec rencontres interdites sont  $\mathcal{NP}$ -difficiles au sens fort.



# Chapitre 4

## Problèmes polynomiaux

### Sommaire

<b>4.1</b>	<b>Introduction</b>	<b>35</b>
<b>4.2</b>	<b>Cas sans rencontres interdites et sans retards des participants</b>	<b>35</b>
<b>4.3</b>	<b>Cas sans rencontres interdites avec retards pour la population la moins nombreuse</b>	<b>36</b>
<b>4.4</b>	<b>Cas sans rencontres interdites et avec retards pour une population</b>	<b>37</b>
<b>4.5</b>	<b>Cas particulier : minimisation de la date de fin globale</b>	<b>41</b>

### 4.1 Introduction

Nous démontrons dans ce chapitre que les cas où aucune rencontre n'est interdite peuvent être résolus en temps polynomial à l'exception du cas où les deux populations peuvent avoir des retards (qui reste ouvert). Toutes ces démonstrations sont inspirées du résultat de complexité du problème d'open shop  $O|r_i, p_{ij} = 1|L_{\max}$  qui est polynomial. En marge de ces résultats, nous présentons la complexité d'un cas particulier du problème de speed-dating pour le critère  $C_{\max}$ .

### 4.2 Cas sans rencontres interdites et sans retards des participants

**Proposition 4.2.1.** *Le problème  $1-1MS|K_{m,n}|W_{\max}$  se résout en  $O(mn)$ . Dans ce cas, la valeur optimale de l'objectif est  $W_{\max} = |m - n|$ .*

*Démonstration.* 4.2.1

Nous rappelons que ce problème est le cas où aucune rencontre n'est interdite et tous les participants arrivent à l'heure.

Sans restreindre la généralité, on suppose que  $m \geq n$ . On construit la solution de la manière suivante. Pour chaque  $j \in \llbracket 0, n - 1 \rrbracket$ , la femme  $F_j$  rencontre l'homme  $M_j$  pendant le premier créneau. Chaque femme rencontre ensuite les hommes dans leur ordre de numérotation. De cette manière, la rencontre entre l'homme  $M_i$  et la femme  $F_j$  a lieu à la date  $(i - j) \bmod m$ . La valeur de l'objectif associé à cette solution est le nombre de créneaux d'attente de l'homme  $M_{m-1}$  qui rencontre la femme  $F_1$  au créneau  $m - 1$ . Ce créneau est le dernier de la soirée après  $n - 1$  rencontres, c'est-à-dire  $W_{\max} = (m - 1) - (n - 1) = m - n$ . Cette solution est optimale car aucune des femmes ne peut finir avant la date  $m$  (qui est le nombre de rencontres de toutes les femmes) et car un homme qui rencontre une femme à la date  $m - 1$  attend nécessairement  $m - n$  créneaux. Le tableau 4.1 illustre cette construction pour une instance à deux femmes et trois hommes. On a bien ici  $W_{\max} = |m - n| = |3 - 2| = 1$ .

□

**Corollaire 4.2.2.** *Le problème  $1-1MS|K_{m,m}|W_{\max}$  se résout en  $O(m^2)$ .*

Participants\Créneaux	0	1	2	$W_i^{(1)}$ ou $W_j^{(2)}$
$M_0$	$F_0$	-	$F_1$	1
$M_1$	$F_1$	$F_0$		0
$M_2$	-	$F_1$	$F_0$	1
$F_0$	$M_0$	$M_1$	$M_2$	0
$F_1$	$M_1$	$M_2$	$M_0$	0

Tableau 4.1 – Solution optimale triviale pour une instance de type  $1-1MS|K_{m,n}|W_{\max}$ .

### 4.3 Cas sans rencontres interdites avec retards pour la population la moins nombreuse

**Proposition 4.3.1.** *Le problème  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  se résout en  $O(mn)$  si  $m \leq n$ . La valeur optimale de l'objectif pour une instance de ce problème est  $W_{\max} = \max_{i \in \llbracket 0, m-1 \rrbracket} \{r_i^{(1)}\} + n - m$ .*

*Démonstration.* 4.3.1

Nous rappelons que le problème auquel nous nous intéressons ici est le cas où les femmes sont à l'heure et sont plus nombreuses que les hommes, les hommes peuvent arriver en retard et aucune rencontre n'est interdite.

Comme présenté dans la section 2.4.1 à la page 24, on peut assimiler le problème de speed-dating avec retards pour les hommes et sans rencontres interdites  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  au problème d'open shop  $O|r_i, p_{ij} = 1|L_{\max}$  dans lequel chaque machine est associée à une femme, chaque job à un homme et chaque opération représente une rencontre. Chaque job a une date de disponibilité égale à la date d'arrivée de l'homme associé.

Dans (Kellerer et al., 1995), les auteurs présentent une solution optimale qu'on peut déterminer en  $O(mn)$  pour le problème dans lequel le nombre de machines (qui correspond au nombre  $n$  de femmes dans notre problème) est au moins égal au nombre de jobs (qui correspond au nombre  $m$  d'hommes dans notre problème). Dans ce cas, il est alors possible de démarrer chaque job dès sa date de disponibilité car il y a plus de machines que de jobs. Ensuite, les opérations d'un job peuvent s'enchaîner sans attentes sur les machines de manière à ce que le job soit effectué en exactement  $n$  unités de temps.

---

**Algorithme 1** Résolution de  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  avec  $n \geq m$

---

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}$  classés par ordre croissant

- 1: **Pour**  $i \in \llbracket 0, m-1 \rrbracket$  **faire**
  - 2:   Choisir une femme  $F_j \in \mathcal{F}$  libre à la date  $r_i^{(1)}$
  - 3:   **Pour**  $k \in \llbracket r_i^{(1)}, r_i^{(1)} + n - 1 \rrbracket$  **faire**
  - 4:     Planifier la rencontre  $(M_i, F_j)$  à la date  $k$
  - 5:      $j := (j + 1) \bmod n$
  - 6:   **Fin Pour**
  - 7: **Fin Pour**
- 

L'algorithme 1 est l'adaptation au problème de speed-dating de l'algorithme proposé dans (Kellerer et al., 1995). Cet algorithme consiste à planifier les rencontres de chaque homme par ordre de dates d'arrivée. Pour chaque homme, une femme est disponible dès sa date d'arrivée car le nombre de femmes est au moins égal au nombre d'hommes. Par construction, aucune femme n'effectue plus d'une rencontre pendant un créneau donné.

Nous illustrons cet algorithme sur l'instance suivante :

- Nombre d'hommes :  $m = 3$ .
- Nombre de femmes :  $n = 4$ .

- Aucune rencontre n'est interdite :  $\forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, p_{ij} = 1$ .
- Dates d'arrivée des hommes :  $r_0^{(1)} = r_1^{(1)} = 0$  et  $r_2^{(1)} = 3$ .
- Dates d'arrivée des femmes :  $r_0^{(2)} = r_1^{(2)} = r_2^{(2)} = r_3^{(2)} = 0$ .

La planification des rencontres sur notre exemple se fait de la manière décrite dans le tableau 4.2.

Participants\Créneaux	0	1	2	3	4	5	6	nb. attentes
$M_0$	$F_0$	$F_1$	$F_2$	$F_3$				0
$M_1$	$F_1$	$F_2$	$F_3$	$F_0$				0
$M_2$				$F_1$	$F_2$	$F_3$	$F_0$	0
$F_0$	$M_0$	-	-	$M_1$	-	-	$M_2$	4
$F_1$	$M_1$	$M_0$	-	$M_2$				1
$F_2$	-	$M_1$	$M_0$	-	$M_2$			2
$F_3$	-	-	$M_1$	$M_0$	-	$M_2$		3

Tableau 4.2 – Solution optimale pour l'instance de  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  avec  $m \leq n$ .

La solution déterminée par l'algorithme 1 est à la fois optimale pour  $W_{\max}^{(1)}$  et  $W_{\max}^{(2)}$ . Les hommes n'ont pas de créneaux d'attente et l'attente maximale d'une femme est  $\max_{i \in \llbracket 0, m-1 \rrbracket} r_i^{(1)} + n - m$ . En effet, il est nécessaire qu'une femme effectue une rencontre à la date  $\max_{i \in \llbracket 0, m-1 \rrbracket} r_i^{(1)} + n - 1$ , qui est la date de la dernière rencontre de l'homme arrivé le plus tard. Cette femme effectue  $m$  rencontres avec les  $m$  hommes présents en terminant sa soirée à la date  $\max_{i \in \llbracket 0, m-1 \rrbracket} \{r_i^{(1)}\} + n$  ce qui démontre notre résultat. On en déduit que  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  se résout en  $O(mn)$  si  $m \leq n$ .  $\square$

**Corollaire 4.3.2.** *Le problème  $1-1MS|K_{m,n}, r_j^{(2)}|W_{\max}$  se résout en  $O(mn)$  si  $m \geq n$ .*

*Démonstration.* 4.3.2

Ce résultat se déduit de la proposition 4.3.1, il suffit pour cela d'inverser le rôle des populations (cf. figure 3.3).  $\square$

**Corollaire 4.3.3.** *Les problèmes  $1-1MS|K_{m,m}, r_i^{(1)}|W_{\max}$  et  $1-1MS|K_{m,m}, r_j^{(2)}|W_{\max}$  se résolvent en  $O(m^2)$ .*

*Démonstration.* 4.3.3

On remarque que les problèmes  $1-1MS|K_{m,m}, r_i^{(1)}|W_{\max}$  et  $1-1MS|K_{m,m}, r_j^{(2)}|W_{\max}$  sont des sous-problèmes de  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  dans lesquels  $m = n$  (cf. figure 3.1a).  $\square$

## 4.4 Cas sans rencontres interdites et avec retards pour une population

Nous démontrons maintenant que le problème sans rencontres interdites et avec retards autorisés pour les hommes reste polynomial si le nombre d'hommes est supérieur au nombre de femmes.

**Proposition 4.4.1.** *On peut résoudre le problème  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$  en  $O(m^{5/2}n^{3/2} \log(mn))$ .*

*Démonstration.* 4.4.1

Ce problème ressemble au problème d'open shop  $O|r_i, p_{ij} = 1|L_{\max}$ . On en déduit l'algorithme 2 qui est une adaptation de l'algorithme de résolution de  $O|r_i, p_{ij} = 1|L_{\max}$  développé dans (Kellerer et al., 1995). Sans restreindre la généralité, on suppose que le premier homme arrive à l'heure et on ordonne les hommes par ordre de date d'arrivée, c'est-à-dire  $0 = r_0^{(1)} \leq r_1^{(1)} \leq \dots \leq r_{n-1}^{(1)}$ .

De la même manière que dans (Kellerer et al., 1995), on suppose que  $\forall k \in \llbracket 0, m-2 \rrbracket, kn > r_{k+1}^{(1)}$ . Sinon, si  $k_0$  est le plus petit élément de  $\llbracket 0, m-2 \rrbracket$  tel que  $k_0 n \leq r_{k_0+1}^{(1)}$ , alors les  $n \times k_0$  rencontres des hommes  $M_0$  à  $M_{k_0}$  peuvent

être planifiées entre les dates  $r_0^{(1)} = 0$  et  $r_{k_0+1}^{(1)}$  et celles des hommes  $k_0 + 1$  à  $m - 1$  ont lieu nécessairement après la date  $r_{k_0+1}^{(1)}$ . On peut donc diviser notre problème en deux sous-problèmes qui peuvent être résolus indépendamment.

L'algorithme 2 se décompose en trois étapes principales. Nous illustrons cet algorithme étape par étape avec l'exemple suivant.

- Nombre d'hommes :  $m = 3$ .
- Nombre de femmes :  $n = 2$ .
- Aucune rencontre n'est interdite :  $\forall (i, j) \in \llbracket 0, m - 1 \rrbracket \times \llbracket 0, n - 1 \rrbracket, p_{ij} = 1$ .
- Dates d'arrivée des hommes :  $r_0^{(1)} = r_1^{(1)} = 0$  et  $r_2^{(1)} = 3$ .
- Dates d'arrivée des femmes :  $r_0^{(2)} = r_1^{(2)} = 0$ .

---

**Algorithme 2** Résolution de  $1-1MS|K_{m,n}, r_i^{(1)}|W_{\max}$

---

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}$

- 1: Déterminer une borne supérieure notée *upperBound* de  $W_{\max}^*$ .
  - 2: Déterminer la valeur de  $W_{\max}^*$  et les créneaux de rencontres des hommes.
  - 3: Affecter les femmes à chaque couple homme-créneau (problème de coloration d'arêtes dans un graphe biparti).
- 

### 1. Détermination d'une borne supérieure de $W_{\max}$

Nous rappelons que  $W_{\max} = \max(W_{\max}^{(1)}, W_{\max}^{(2)})$ . Dans le cas où aucune rencontre n'est interdite, on a :

- $W_{\max}^{(1)} = \max_{i \in \llbracket 0, m-1 \rrbracket} \{C_i^{(1)} - (r_i^{(1)} + n)\} = \max_{i \in \llbracket 0, m-1 \rrbracket} \{C_i^{(1)} - r_i^{(1)}\} - n$  et
- $W_{\max}^{(2)} = \max_{j \in \llbracket 0, n-1 \rrbracket} \{C_j^{(2)} - m\} = \max_{j \in \llbracket 0, n-1 \rrbracket} \{C_j^{(2)}\} - m$  car toutes les femmes sont à l'heure.

Une borne supérieure triviale de  $W_{\max}$  est obtenue en affectant les rencontres de chaque homme  $M_i \in \mathcal{M}$  entre la date  $(i - 1)n$  et la date  $i \times n$  ce qui est possible car  $\forall j \in \llbracket 2, n \rrbracket, r_i^{(1)} < (j - 1)n$ . Une telle solution est présentée pour notre exemple dans le tableau 4.3. Dans cette configuration, la soirée se termine à la date  $mn$  d'où  $\max_{F_j \in \mathcal{F}} \{C_j^{(2)}\} = \max_{M_i \in \mathcal{M}} \{C_i^{(1)}\} = mn$ . L'attente maximum des femmes est donc  $W_{\max}^{(2)} = mn - m$

et l'attente maximum des hommes  $W_{\max}^{(1)}$  est bornée par  $mn - n$ . Dans tous les cas,  $mn - m \leq mn - n$  car  $n \leq m$ . Une borne supérieure de  $W_{\max}$  est donc  $upperBound = \max(W_{\max}^{(1)}, mn - m) \leq mn - n$ .

Dans notre exemple, la solution triviale du tableau 4.3 permet d'obtenir la borne supérieure :  $upperBound = 3$ .

Participants\Créneaux	0	1	2	3	4	5	Nb Attentes
$M_0$	$F_0$	$F_1$					0
$M_1$	-	-	$F_0$	$F_1$			2
$M_2$				-	$F_0$	$F_1$	1
$F_0$	$M_0$	-	$M_1$	-	$M_2$		2
$F_1$	-	$M_0$	-	$M_1$	-	$M_2$	3

Tableau 4.3 – Solution triviale pour l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max}$ .

### 2. Détermination des créneaux des hommes et de la valeur optimale de $W_{\max}$ .

On recherche ensuite par dichotomie la valeur optimale de  $W_{\max}$ . Dans cette étape, on décrit la procédure permettant de déterminer s'il existe une solution vérifiant  $W_{\max} \leq k$  où  $k \in \mathbb{N}$ . Cette recherche se fait de la manière décrite dans l'algorithme 3.

---

**Algorithme 3** Détermination de la valeur de  $W_{\max}^*$  et des créneaux de rencontre des hommes
 

---

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in [0, m-1]}, upperBound$

```

1:  $lb := 0$ 
2:  $ub := upperBound$ 
3: Tant que  $ub \neq lb$  faire
4:    $k := \lceil \frac{ub+lb}{2} \rceil$ 
5:   Déterminer s'il existe une solution vérifiant  $W_{\max} \leq k$ .
6:   Si oui Alors
7:      $ub := k$ 
8:   Sinon
9:      $lb := k + 1$ 
10:  Fin Si
11: Fin Tant que
12:  $W_{\max}^* := ub$ 
13: Déterminer les créneaux de rencontre de chaque homme.
```

---

Le test de la ligne 5 de l'algorithme 3 s'effectue par un algorithme de recherche de flot maximum dans un graphe orienté (par exemple par l'algorithme de Fernández-Baca et Martel (Fernández-Baca et Martel, 1989) spécifique au flot dans les graphes bipartis). Ce graphe orienté valué est noté  $G = (\{s\} \cup \mathcal{M} \cup \mathcal{S} \cup \{t\}, A, \mu)$  où  $s$  et  $t$  sont respectivement le sommet source et le sommet puits ;  $\mathcal{M}$  représente l'ensemble des hommes et  $\mathcal{S}$  représente l'ensemble des créneaux disponibles pour la soirée. On a  $|\mathcal{M}| = m$  et  $|\mathcal{S}| = k + m$  car  $W_{\max}^{(2)} = \max_{j \in [0, n-1]} \{C_j^{(2)}\} - m \leq W_{\max} \leq k$  (on cherche une solution dans laquelle le nombre maximal de créneaux d'attente est inférieur ou égal à  $k$ ). L'unique différence avec l'algorithme de (Kellerer et al., 1995) intervient dans le choix de  $\mathcal{S}$  : en effet, Kellerer et al. ne restreignent pas le nombre total de créneaux disponibles. Il n'est pas nécessaire dans le cas de l'open shop de borner le nombre de créneaux car l'objectif est de minimiser les retards des jobs ; dans notre cas, on minimise les retards des femmes et des hommes. On crée un arc de capacité  $n$  reliant  $s$  vers chacun des sommets de  $\mathcal{M}$ . De même, on crée un arc de capacité  $n$  d'origine chacun des sommets de  $\mathcal{S}$  et de destination  $t$ . On définit un arc de capacité 1 d'origine un homme de  $\mathcal{M}$  et de destination un sommet créneau de  $\mathcal{S}$  si et seulement si cet homme peut effectuer une rencontre pendant ce créneau sans que son nombre de créneaux d'attente dépasse  $k$ , c'est-à-dire :  $\forall (M_i, d) \in \mathcal{M} \times \mathcal{S}, ((M_i, d) \in A \iff d \in [r_i^{(1)}, \min(r_i^{(1)} + n + k - 1, k + m)])$ .

Dans notre exemple, lors de la première itération de la boucle **Tant que**,  $lb = 0$  et  $ub = 3$  ; on cherche s'il existe une solution au problème pour  $W_{\max} \leq \lceil 3/2 \rceil = 2$ . On souhaite que  $j \in [0, n-1], W_j^{(2)} \leq 2$ . La soirée est donc limitée à cinq créneaux (de  $S_0$  à  $S_4$ ). En effet, si une femme effectue une rencontre à l'issue du créneau  $S_4$ , elle aura nécessairement trois créneaux d'attente. Chaque homme  $M_i$  pour  $i \in \{0, 1, 2\}$  peut effectuer des rencontres entre les dates  $r_i^{(1)}$  et  $r_i^{(1)} + n + k = r_i^{(1)} + 4$  car  $k = 2$  et  $n = 2$ . Le graphe de flot associé est présenté dans la figure 4.1a.

On applique ensuite un algorithme de flot maximum à ce graphe (Fernández-Baca et Martel, 1989). Si la valeur maximale du flot est  $mn$ , alors  $W_{\max} \leq k$ . Les arcs qui relient un sommet de  $\mathcal{M}$  à un sommet de  $\mathcal{S}$  et traversés par un flux de 1 correspondent à l'affectation des hommes aux créneaux de leurs rencontres. Cette étape correspond à la ligne 13 de l'algorithme 3.

Dans notre exemple, pour qu'il existe une solution telle que  $W_{\max} \leq 2$ , il suffit que le flot maximal soit de 6 (nombre de rencontres). Dans cet exemple, il existe un flot de valeur 6 qui est représenté sur la figure 4.1b.



Les hommes  $M_0$ ,  $M_1$  et  $M_2$  peuvent effectuer leurs rencontres lors des créneaux  $S_0$  et  $S_1$  ;  $S_2$  et  $S_3$  ;  $S_3$  et  $S_4$  respectivement.

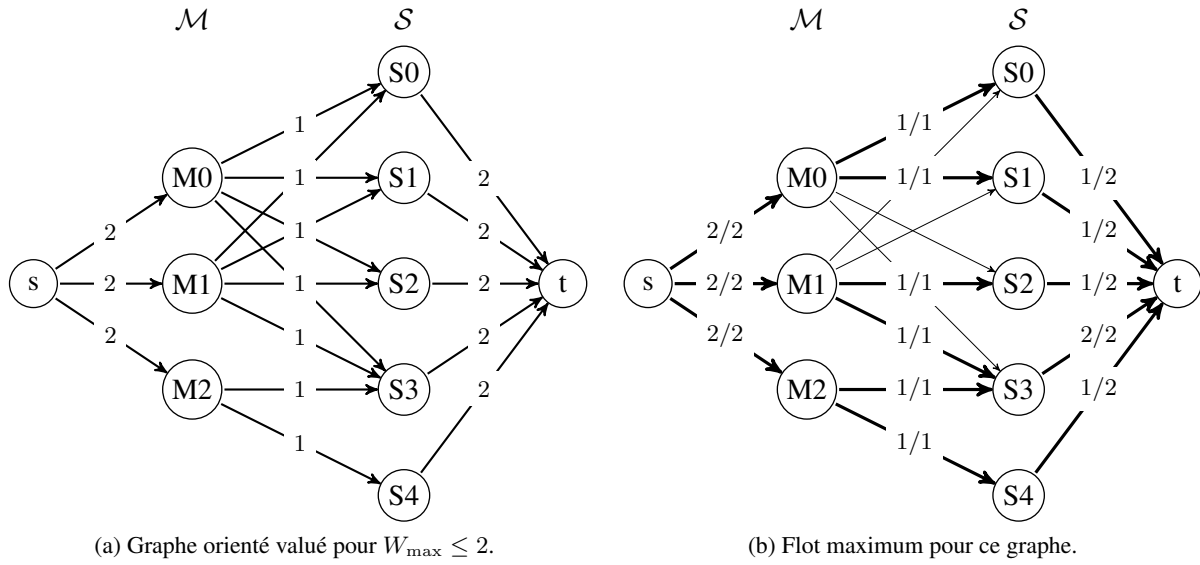


FIGURE 4.1 – Affectation des hommes aux créneaux de la soirée.

Il existe une solution pour  $W_{\max} \leq 2$ . On affecte donc 2 à la borne supérieure  $ub$ . Lors de la deuxième itération de la boucle **Tant que**, on détermine s'il existe une solution telle que l'objectif  $W_{\max} \leq 1$ . La réponse est non car le graphe de la figure 4.2 (obtenu comme précédemment) n'admet pas de flot maximum de valeur 6 (nombre total de rencontres). En effet, l'homme  $M_2$  ne peut pas effectuer ses deux rencontres : il n'a qu'un créneau disponible et si on lui autorise une rencontre au créneau  $S_4$ , une femme aura un nombre de créneaux d'attente égal à 2.

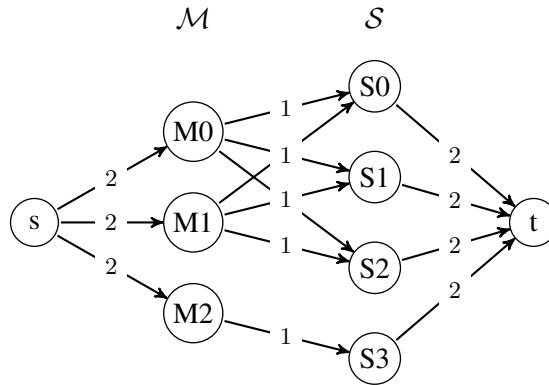


FIGURE 4.2 – Graphe orienté valué pour  $W_{\max} \leq 1$ .

### 3. Affectation des femmes à couple homme-créneau.

Une fois qu'on a obtenu les créneaux de chaque homme pour la valeur optimale de  $W_{\max}$ , il reste à affecter une femme à chaque couple homme-créneau de manière à ce qu'un homme ne rencontre pas deux fois la même femme et qu'une femme n'ait pas plus d'une rencontre par créneau. On utilise un algorithme de coloration d'arêtes sur le graphe biparti construit à partir des sommets de  $\mathcal{M}$  et  $\mathcal{S}$  et des arcs entre ces sommets dont la valeur du flot est 1. D'après le théorème de König (König, 1916), il existe une coloration de ce graphe avec  $n$  couleurs car le graphe est biparti et que le degré maximal des sommets est exactement  $n$ . On utilise l'algorithme de coloration de Cole et Hopcroft en  $O(E \log V)$  (Cole et Hopcroft, 1982).

Dans notre exemple, le graphe que l'on souhaite colorer est celui de la figure 4.3. On a représenté sur celui-ci une coloration représentant les femmes.

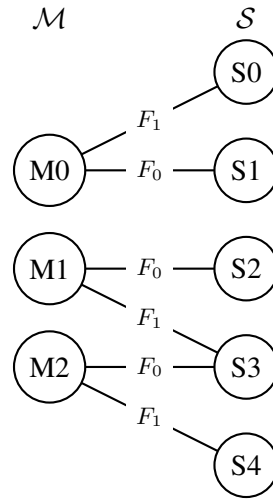


FIGURE 4.3 – Affectation des femmes aux couples hommes-créneaux.

La solution optimale associée est donnée dans le tableau 4.4.

Participants\Créneaux	0	1	2	3	4	$W_i^{(1)}$
$M_0$	$F_1$	$F_0$				0
$M_1$	-	-	$F_0$	$F_1$		2
$M_2$				$F_0$	$F_1$	0
						$W_i^{(1)}$
$F_0$	-	$M_0$	$M_1$	$M_2$		1
$F_1$	$M_0$	-	-	$M_1$	$M_2$	2

Tableau 4.4 – Solution optimale pour l'exemple

La complexité de cet algorithme est équivalent à celle de l'algorithme proposé dans (Kellerer et al., 1995). Celui-ci est en  $O(n^{5/2}m^{3/2}\log(mn))$  où  $n$  et  $m$  désignent respectivement le nombre de jobs et le nombre de machines. On en déduit notre résultat.  $\square$

**Corollaire 4.4.2.** *Le problème  $1-1MS|K_{m,n}, r_j^{(2)}|W_{\max}$  se résout en  $O(n^{5/2}m^{3/2}\log(mn))$ .*

*Démonstration.* 4.4.2

Il suffit pour cela d'appliquer l'algorithme de la proposition 4.4.1 en inversant le rôle des hommes et des femmes (cf. figure 3.3).  $\square$

## 4.5 Cas particulier : minimisation de la date de fin globale

Dans le chapitre 2, nous avons établi le lien entre le problème d'ordonnancement par fournées avec incompatibilités entre les jobs et le problème de speed-dating dans la section 2.4.3. Pour rappel, le problème  $1|p\text{-batch}, G = (V, E), b \geq n, p_i = 1|C_{\max}$  est équivalent au problème de speed-dating  $1-1MS|(U \cup V, E)|C_{\max}$  dans lequel des rencontres sont interdites et tous les participants sont à l'heure. On peut en déduire le résultat suivant :

**Proposition 4.5.1.** *Le problème  $1-1MS|(U \cup V, E)|C_{\max}$  est polynomial.*

*Démonstration.* 4.5.1 Le problème  $1-1MS|(U \cup V, E)|C_{\max}$  est un sous-cas du problème d'ordonnancement par fournées avec incompatibilités entre les jobs dont le graphe de compatibilités entre les rencontres est le graphe de ligne d'un graphe biparti. Le graphe de ligne noté  $L(G)$  d'un graphe  $G$  est obtenu en associant un sommet à chaque

arête de  $G$  et en reliant deux sommets dans  $L(G)$  si et seulement si les arêtes correspondantes dans  $G$  ont une extrémité commune. On vérifie facilement que ce graphe est parfait à l'aide de la caractérisation des graphes parfaits de (Chudnovsky et al., 2006). D'après (Grötschel et al., 1981), le problème de partition en cliques d'un graphe parfait se résout en temps polynomial. Ce problème étant équivalent au problème de speed-dating  $1-1MS|(U \cup V, E)|C_{\max}$ , ce dernier est donc polynomial.  $\square$

## Conclusion

À l'issue de ce chapitre, nous avons établi les résultats de complexité résumés dans le tableau 4.5. Nous avons démontré que si une population est en retard et qu'il n'y a pas de rencontres interdites, le problème est polynomial. Ces résultats sont déduits de problèmes d'open shop avec minimisation du retard algébrique. Le cas avec retards pour les deux populations reste ouvert, et cela constitue une des premières pistes que nous souhaitons poursuivre. Il ne semble pas probable que ce cas se déduise facilement d'un problème d'open shop car les problèmes d'open shop existants ne prennent en compte des retards que pour les jobs. À notre connaissance, il n'y a pas de travaux sur des problèmes d'ordonnancement quelconques considérant des dates de disponibilité pour les machines.

	$\emptyset$	$r_i^{(1)}$	$r_j^{(2)}$	$r_i^{(1)}, r_j^{(2)}$
$K_{m,m}$	$O(m^2)$	$O(m^2)$	$O(m^2)$	
$K_{m,n}$	$O(mn)$	$O(m^{5/2}n^{3/2}\log(mn))$	$O(n^{5/2}m^{3/2}\log(mn))$	

Tableau 4.5 – Problèmes de speed-dating décidables en temps polynomial pour  $W_{\max}$ .

Dans le chapitre suivant, nous étudions les cas de notre problème avec rencontres interdites dont nous allons démontrer la  $\mathcal{NP}$ -difficulté.

# Chapitre 5

## Problèmes $\mathcal{NP}$ -difficiles

### Sommaire

<b>5.1</b>	<b>Introduction</b>	<b>43</b>
<b>5.2</b>	<b>Cas avec rencontres interdites et retards autorisés</b>	<b>43</b>
<b>5.3</b>	<b>Cas avec rencontres interdites et retards pour une population</b>	<b>46</b>
<b>5.4</b>	<b>Cas avec rencontres interdites et sans retards</b>	<b>48</b>
<b>5.5</b>	<b><math>\mathcal{NP}</math>-difficulté au sens fort</b>	<b>51</b>
<b>5.6</b>	<b>Non approximabilité</b>	<b>52</b>

### 5.1 Introduction

Dans ce chapitre, nous établissons la  $\mathcal{NP}$ -complétude de plusieurs problèmes de décision associés à des problèmes de speed-dating. Nous démontrons que tous les cas où des rencontres sont interdites sont  $\mathcal{NP}$ -difficiles au sens fort. Dans un premier temps, on démontre que les cas  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max}$  et  $1-1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|W_{\max}$  sont  $\mathcal{NP}$ -difficiles par réduction partant d'un problème de coloration d'arêtes dans un graphe biparti. Ensuite, on établit par réductions successives que les cas avec retards pour une population puis sans retards sont également  $\mathcal{NP}$ -difficiles au sens fort.

On introduit les notations suivantes pour désigner les problèmes de décision associés aux problèmes d'optimisation de speed-dating. Pour  $\beta \in \{|U| = |V|, r_i^{(1)}, r_j^{(2)}\}$ , on note :

$$1-1MS|(U \cup V, E), \beta|W_{\max} \leq k$$

INSTANCE : Ensembles d'hommes et de femmes  $\mathcal{M}, \mathcal{F}$ , listes de dates d'arrivée des participants  $(r_i^{(1)})_{i \in [0, m-1]}$  et  $(r_j^{(2)})_{j \in [0, n-1]}$  et matrice de rencontres  $(p_{ij})_{(i,j) \in [0, m-1] \times [0, n-1]}$ .

SOLUTION : Ordonnancement réalisable des rencontres respectant les contraintes de dates d'arrivée et de rencontres interdites tel que  $W_{\max} \leq k$ .

### 5.2 Cas avec rencontres interdites et retards autorisés

**Proposition 5.2.1.** *Le problème  $1-1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|W_{\max}$  est  $\mathcal{NP}$ -difficile.*

*Démonstration.* 5.2.1

En premier lieu, nous démontrons que le problème  $1-1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  est  $\mathcal{NP}$ -complet en effectuant une réduction polynomiale à partir d'un problème  $\mathcal{NP}$ -complet. Pour cela nous introduisons ci-après une généralisation du problème de coloration des arêtes d'un graphe.

Le problème d'extension de précoloration des arêtes d'un graphe qu'on notera EPREXT (Edge PRecoloration EXTension) est une généralisation du problème classique de  $k$ -coloration ( $k \in \mathbb{N}$ ) des arêtes d'un graphe. Ce dernier consiste à affecter une couleur à chaque arête d'un graphe de manière à ce que deux arêtes adjacentes aient des

couleurs différentes et que le nombre total de couleurs utilisées soit inférieur ou égal à  $k$ . La coloration des arêtes d'un graphe  $G = (V, E)$  est représentée par une application de coloration  $c : E \rightarrow \llbracket 0, k-1 \rrbracket$  où  $c(e)$  représente la couleur de l'arête  $e$ . Le problème EPREXT prend en entrée un graphe  $G = (V, E)$ , un sous-ensemble  $D \subseteq E$ , un entier naturel  $k$  et une application de précoloration  $f : D \rightarrow \llbracket 0, k-1 \rrbracket$ . Ce problème consiste à décider s'il existe une  $k$ -coloration  $c$  de  $G$  vérifiant  $\forall e \in D, c(e) = f(e)$ . On note  $\text{EPREXT}K_{\lambda, \lambda}$  le problème de décision suivant.

$\text{EPREXT}K_{\lambda, \lambda}$

INSTANCE : Graphe biparti complet de taille  $\lambda$  noté  $K_{\lambda, \lambda} = (U \cup V, E)$  avec  $|U| = |V| = \lambda$ ; un sous-ensemble  $D$  de  $E$ ; une application de précoloration  $f : D \rightarrow \llbracket 0, \lambda-1 \rrbracket$ .

SOLUTION : Une coloration de  $K_{\lambda, \lambda}$  notée  $c : E \rightarrow \llbracket 0, \lambda-1 \rrbracket$  telle que deux arêtes adjacentes aient deux couleurs différentes respectant la précoloration  $f$ .

Le problème  $\text{EPREXT}K_{\lambda, \lambda}$  est  $\mathcal{NP}$ -complet (Colbourn, 1984).

1. Il est clair que  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|_{W_{\max}} = 0 \in \mathcal{NP}$  : une solution OUI est vérifiable en temps polynomial. Il suffit de vérifier que chaque participant enchaîne ses rencontres sans attentes à partir de sa date de disponibilité et que toutes les contraintes du problème sont vérifiées.

2. **Construction en temps polynomial d'une instance de  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|_{W_{\max}} = 0$  à partir d'une instance de  $\text{EPREXT}K_{\lambda, \lambda}$ .**

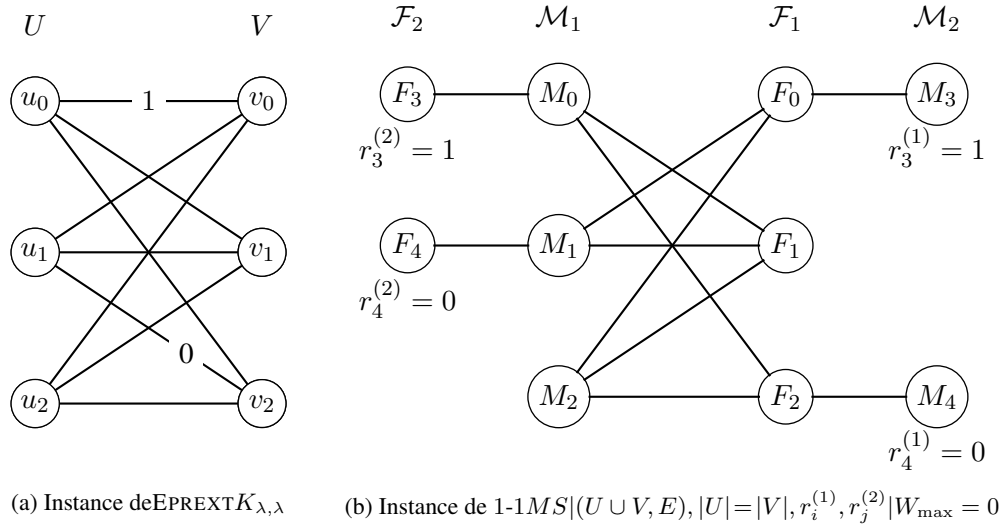
Soit  $\lambda$  un entier naturel non nul et  $f : D \subseteq E \rightarrow \llbracket 0, \lambda-1 \rrbracket$  l'application de précoloration d'un graphe biparti complet de taille  $\lambda$  :  $K_{\lambda, \lambda} = (U \cup V, E)$ . À chaque sommet  $u_i$  de  $U = \{u_0, \dots, u_{\lambda-1}\}$ , on associe l'homme  $M_i$  et à chaque sommet  $v_j$  de  $V = \{v_0, \dots, v_{\lambda-1}\}$ , on associe la femme  $F_j$ . On note  $\mathcal{M}_1 = \{M_i \mid u_i \in U\}$  et  $\mathcal{F}_1 = \{F_j \mid v_j \in V\}$ . Ces participants ont pour date de disponibilité 0, c'est-à-dire qu'ils n'ont pas de retard. Une rencontre entre un homme  $M_i \in \mathcal{M}_1$  et une femme  $F_j \in \mathcal{F}_1$  a lieu si et seulement si l'arête  $[u_i, v_j]$  de  $K_{\lambda, \lambda}$  n'est pas colorée. Formellement,  $\forall u_i \in U, \forall v_j \in V, p_{ij} = 1 \iff [u_i, v_j] \notin D$ . Pour chaque arête précolorée  $[u_i, v_j] \in D$ , on ajoute une femme qui n'a qu'une seule rencontre avec l'homme  $M_i$  et un homme qui n'a qu'une seule rencontre avec la femme  $F_j$ . La date de disponibilité de ces deux participants est  $f([u_i, v_j])$ , c'est à dire qu'elle est liée à la couleur de l'arête  $[u_i, v_j]$  dans  $K_{\lambda, \lambda}$ . Les ensembles de ces femmes et hommes supplémentaires sont notés  $\mathcal{F}_2$  et  $\mathcal{M}_2$  respectivement. Ainsi,  $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$  et  $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ . Par définition du graphe  $K_{\lambda, \lambda}$ ,  $|\mathcal{M}_1| = |\mathcal{F}_1|$ . On remarque également que  $|\mathcal{M}_2| = |\mathcal{F}_2|$ . Il y a donc autant d'hommes que de femmes. Cette construction se fait clairement en temps polynomial car il y a au maximum  $2(\lambda + \lambda^2)$  participants ( $2\lambda$  participants correspondent aux sommets de  $K_{\lambda, \lambda}$  et au plus  $2\lambda^2$  participants sont associés aux arêtes précolorées).

La figure 5.1 illustre la construction que nous décrivons. La figure 5.1a représente une instance de  $\text{EPREXT}K_{\lambda, \lambda}$ , la précoloration est indiquée par les nombres notés sur les arêtes. La figure 5.1b représente le graphe des rencontres du problème de speed-dating associé. Dans ce graphe, les sommets représentent les participants et les arêtes les rencontres à effectuer. Les dates de disponibilité des participants sont indiquées sous chaque sommet. L'absence de date de disponibilité d'un sommet indique que ce participant arrive à la date 0.

Ici, l'arête  $[u_0, v_0]$  du problème de coloration d'arêtes est précolorée par 1. On ajoute donc les participants  $M_3$  et  $F_3$  qui rencontrent respectivement  $F_0$  et  $M_0$ , ceux-ci ont pour date de disponibilité 1. De même, l'arête  $[u_1, v_2]$  a pour précoloration 0, les participants  $M_4$  et  $F_4$  ajoutés rencontrent respectivement  $F_2$  et  $M_1$  et ont pour date de disponibilité 0.

3. **Si  $\text{EPREXT}K_{\lambda, \lambda}$  a une solution alors  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|_{W_{\max}} = 0$  a une solution.**  
On suppose que l'instance d' $\text{EPREXT}K_{\lambda, \lambda}$  admet une solution, on en déduit la suivante pour l'instance de  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)}|_{W_{\max}} = 0$  associée :

- quels que soient  $M_i \in \mathcal{M}_1$  et  $F_j \in \mathcal{F}_1$  tels que  $p_{ij} = 1$ , la rencontre entre l'homme  $M_i$  et la femme  $F_j$  a lieu à la date  $c([u_i, v_j])$  ;

FIGURE 5.1 –  $\text{EPREXT}_{\lambda, \lambda} \propto 1\text{-}1\text{MS}[(U \cup V, E), r_i^{(1)}, r_j^{(2)} | W_{\max} = 0$ 

- quels que soient  $M_i \in \mathcal{M}_1$  et  $F_j \in \mathcal{F}_2$  tels que  $p_{ij} = 1$ , la rencontre entre l'homme  $M_i$  et la femme  $F_j$  a lieu à la date  $r_j^{(2)}$  ;
- quels que soient  $M_i \in \mathcal{M}_2$  et  $F_j \in \mathcal{F}_1$  tels que  $p_{ij} = 1$ , la rencontre entre l'homme  $M_i$  et la femme  $F_j$  a lieu à la date  $r_i^{(1)}$  .

Cet ordonnancement des rencontres est valide car les participants de  $\mathcal{M}_1 \cup \mathcal{F}_1$  ont exactement  $\lambda$  rencontres à effectuer sur  $\lambda$  créneaux, ils n'ont donc pas de créneaux d'attente. Par ailleurs, leurs rencontres s'effectuent lors de créneaux différents ce qui est dû à la coloration des arêtes de  $K_{\lambda, \lambda}$ . Les participants de  $\mathcal{M}_2 \cup \mathcal{F}_2$  effectuent leur unique rencontre dès leur date de disponibilité ce qui implique qu'ils n'ont pas d'attente, d'où  $W_{\max} = 0$ .

**4. Si une solution existe pour  $1\text{-}1\text{MS}[(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)} | W_{\max} = 0$  alors  $\text{EPREXT}_{\lambda, \lambda}$  a une solution.**

Si le problème de speed-dating admet une solution avec  $W_{\max} = 0$ , le nombre de créneaux nécessaires à la soirée est exactement  $\lambda$  car, par construction, les participants de  $\mathcal{M}_1 \cup \mathcal{F}_1$  ont chacun  $\lambda$  rencontres à effectuer. En effet, chaque participant de  $\mathcal{F}_1 \cup \mathcal{M}_1$  arrive à l'heure et provient d'un sommet du graphe biparti complet  $K_{\lambda, \lambda}$ , ce sommet a donc pour degré  $\lambda$  et pour chaque arête incidente à ce sommet, une rencontre est associée au participant. À chaque arête non colorée de  $K_{\lambda, \lambda}$ , on associe la date du créneau de la rencontre correspondante. Un sommet de  $K_{\lambda, \lambda}$  n'a pas deux arêtes adjacentes de même couleur car le participant associé à ce sommet n'a pas deux rencontres pendant le même créneau. Cette coloration est compatible avec la précoloration car si un participant correspond à un sommet dont une arête adjacente est précolorée, celui-ci a une rencontre obligatoire au créneau correspondant à la précoloration avec un participant de  $\mathcal{M}_2 \cup \mathcal{F}_2$ .

Ainsi, nous avons démontré que  $1\text{-}1\text{MS}[(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)} | W_{\max} = 0$  est  $\mathcal{NP}$ -complet. On en déduit que le cas plus général  $1\text{-}1\text{MS}[(U \cup V, E), |U| = |V|, r_i^{(1)}, r_j^{(2)} | W_{\max} \leq k$  est également  $\mathcal{NP}$ -complet et que le problème d'optimisation associé à celui-ci est  $\mathcal{NP}$ -difficile.  $\square$

**Corollaire 5.2.2.** *Le problème plus général  $1\text{-}1\text{MS}[(U \cup V, E), r_i^{(1)}, r_j^{(2)} | W_{\max}$  dans lequel les nombres d'hommes et de femmes sont quelconques est  $\mathcal{NP}$ -difficile.*

*Démonstration.* 5.2.2

Ce résultat se déduit de la proposition 5.2.1 et du graphe de la figure 3.1a.  $\square$

### 5.3 Cas avec rencontres interdites et retards pour une population

Nous démontrons maintenant que les problèmes dans lesquels des rencontres sont interdites et des retards ne sont autorisés que pour une seule population restent  $\mathcal{NP}$ -difficiles quelle que soit la proportion entre hommes et femmes.

Le lemme suivant permet d'établir que l'ensemble des instances de  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max}$  dont l'objectif optimal est nul ont leur ensemble de dates de disponibilité  $\{r_i^{(1)} \mid i \in \llbracket 0, m-1 \rrbracket\} \cup \{r_j^{(2)} \mid j \in \llbracket 0, n-1 \rrbracket\}$  borné polynomialement en  $m$  et  $n$ . Ce lemme sera utile dans les démonstrations suivantes.

**Lemme 5.3.1.** *Si une instance du problème  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  pour laquelle  $\min\{r_i^{(1)}, r_j^{(2)} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket\} = 0$  admet une solution, alors les dates de disponibilité des participants sont bornées par un polynôme en  $m$  et  $n$ .*

*Démonstration.* Lemme 5.3.1

Soit une instance de  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  ayant une solution réalisable. On rappelle que le graphe biparti des rencontres de cette instance est  $G = (\mathcal{M} \cup \mathcal{F}, E)$  où  $E = \{[M_i, F_j] \in \mathcal{M} \times \mathcal{F} \mid p_{ij} = 1\}$ . Les rencontres des différentes composantes connexes du graphe sont indépendantes, on peut donc supposer sans perdre de généralité que le graphe est connexe. Rappelons qu'on peut supposer qu'il existe au minimum une femme et un homme arrivant à la date zéro sans restreindre la généralité car dans le cas contraire, on peut décaler la date de début de la soirée à la première date à laquelle sont disponibles deux participants. Comme aucun participant ne peut avoir de créneaux d'attente, une rencontre a lieu à la date 0. On note  $(M_{i^*}, F_{j^*}) \in \mathcal{M} \times \mathcal{F}$  un couple effectuant cette rencontre. Alors,  $r_{i^*}^{(1)} = r_{j^*}^{(2)} = 0$ .

Quel que soit le couple  $(M_i, F_j) \in \mathcal{M} \times \mathcal{F}$  devant se rencontrer,  $r_i^{(1)} < m + r_j^{(2)}$ . En effet, la femme  $F_j$  a au maximum  $m$  rencontres à effectuer ; si l'homme  $M_i$  arrive après la date  $m + r_j^{(2)}$ , la femme  $F_j$  aura nécessairement un créneau d'attente (ou plus) et l'instance n'aura pas de solution. De même,  $r_j^{(2)} < n + r_i^{(1)}$ . De manière formelle :

$$\forall (i, j) \in \llbracket 1, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, p_{ij} = 1 \Rightarrow (r_i^{(1)} < m + r_j^{(2)}) \text{ et } (r_j^{(2)} < n + r_i^{(1)}). \quad (5.1)$$

Soit une femme  $F_{j'} \in \mathcal{F}$ . Il existe une chaîne reliant le sommet  $F_{j^*}$  à  $F_{j'}$  car le graphe  $G$  est connexe. Une telle chaîne peut s'écrire  $[F_{j_1} := F_{j^*}, M_{i_1}, F_{j_2}, \dots, F_{j_k}, M_{i_k}, F_{j'}]$  avec  $i_\lambda \in \llbracket 0, m-1 \rrbracket$  et  $j_\mu \in \llbracket 0, n-1 \rrbracket$  pour  $\lambda, \mu \in \llbracket 1, k \rrbracket$  car le graphe est biparti. D'après les inégalités (5.1),  $r_{j'}^{(2)} < n + r_{i_k}^{(1)} < n + (m + r_{j_k}^{(2)}) < \dots < km + kn + r_{j_1}^{(2)}$  d'où  $r_{j'}^{(2)} < k(m + n)$  puisque  $r_{j_1}^{(2)} = 0$ . Dans le pire des cas, la chaîne passe par tous les sommets de  $\mathcal{F}$  donc  $k \leq n-1$ . On a donc :

$$\forall j \in \llbracket 0, n-1 \rrbracket, r_j^{(2)} < (n-1)(m+n). \quad (5.2)$$

De la même manière, on démontre que :

$$\forall i \in \llbracket 0, m-1 \rrbracket, r_i^{(1)} < (m-1)(m+n). \quad (5.3)$$

□

**Proposition 5.3.2.** *Le problème  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max}$  est  $\mathcal{NP}$ -difficile.*

*Démonstration.* 5.3.2

On effectue une réduction polynomiale de  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  qui est  $\mathcal{NP}$ -complet d'après le corollaire 5.2.2 vers le problème  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ . Nous illustrons cette réduction sur l'exemple à trois femmes et deux hommes dont les données sont les suivantes :

- les dates d'arrivée des hommes  $r_0^{(1)}$  et  $r_1^{(1)}$  sont quelconques ;
- les dates d'arrivée des femmes sont  $r_0^{(2)} = 0$  et  $r_1^{(2)} = r_2^{(2)} = 2$  ;

- la matrice des rencontres est la suivante :

$$p_{ij} = \begin{cases} 1 & \text{si } (i, j) \in \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 2)\}, \\ 0 & \text{sinon.} \end{cases}$$

- Il est clair que  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0 \in \mathcal{NP}$  : une solution OUI est vérifiable en temps polynomial. Il suffit de vérifier que chaque participant enchaîne les rencontres qu'il doit effectuer sans attente à partir de sa date de disponibilité et que toutes les autres contraintes sont vérifiées.
- Construction en temps polynomial d'une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  à partir d'une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$ .**

Partant d'une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$ , on construit une nouvelle instance de speed-dating avec des dates de disponibilité nulles pour les femmes.

Pour cela, pour chaque femme  $F_j \in \mathcal{F}$ , on ajoute un ensemble  $\mathcal{M}_{F_j}$  de  $r_j^{(2)}$  hommes. On note  $\mathcal{M}^+ = \bigcup_{j \in \llbracket 0, n-1 \rrbracket} \mathcal{M}_{F_j}$  de cardinal  $|\mathcal{M}^+| = m'$ . Dans cette nouvelle instance, toute femme de  $\mathcal{F}$  reste inchangée tandis que l'ensemble des hommes devient  $\mathcal{M} \cup \mathcal{M}^+$ . Chaque rencontre de l'instance de départ est conservée et chaque homme de  $\mathcal{M}_{F_j}$  (pour  $j \in \llbracket 0, n-1 \rrbracket$ ) n'effectue qu'une rencontre avec la femme  $F_j$ . La matrice des rencontres  $(P_{ij})_{(i,j) \in \llbracket 0, m+m'-1 \rrbracket \times \llbracket 0, n-1 \rrbracket}$  est ainsi définie de la manière suivante :

$$\forall j \in \llbracket 0, n-1 \rrbracket, \quad P_{ij} = \begin{cases} p_{ij} & \text{si } i \in \llbracket 0, m-1 \rrbracket; \\ 1 & \text{si } M_i \in \mathcal{M}_{F_j}; \\ 0 & \text{sinon.} \end{cases}$$

Dans la nouvelle instance, les dates d'arrivée des participants de  $\mathcal{M} \cup \mathcal{F}$  sont conservées et pour chaque  $j \in \llbracket 0, n-1 \rrbracket$ , le  $k$ -ème homme de  $\mathcal{M}_{F_j}$  arrive à la date  $k-1$ .

La figure 5.2 illustre cette construction sur notre exemple. Le graphe de gauche représente l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  et celui de droite représente l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  construite à partir de la première. Ainsi les hommes  $M_2$  et  $M_3$  de notre exemple ont pour date d'arrivée respective 0 et 1.

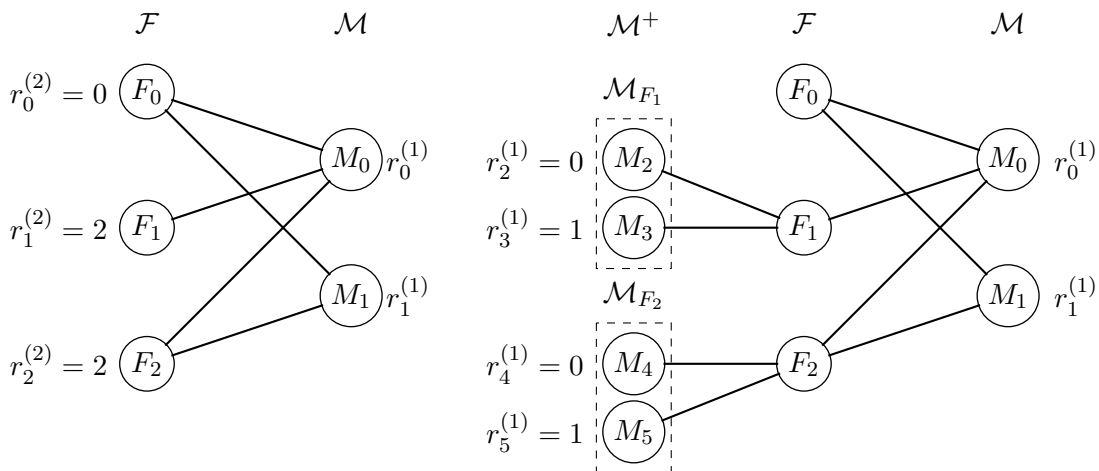


FIGURE 5.2 – Réduction  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0 \propto 1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ .

Dans le cas général, cette construction ne se fait pas en temps polynomial car on ajoute  $m'$  participants, avec  $m' = \sum_{j \in \llbracket 0, n-1 \rrbracket} r_j^{(2)}$  qui peut être arbitrairement grand. Cependant, si  $\sum_{j \in \llbracket 0, n-1 \rrbracket} r_j^{(2)}$  n'est pas bornée polynomialement en  $m$  et  $n$ , alors l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  n'admet trivialement pas



de solution d'après le lemme 5.3.1. On peut donc supposer sans perte de généralité que cette construction se fait à partir d'une instance telle que  $m'$  soit borné polynomialement. La construction est alors bien polynomiale.

3. **Si une solution existe pour  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  alors  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  admet une solution.**

En effet, il suffit de conserver les créneaux des rencontres entre les participants de  $\mathcal{F}$  et de  $\mathcal{M}$ . Ceux-ci sont compatibles avec les créneaux des rencontres de chaque femme  $F_j \in \mathcal{F}$  avec les hommes de  $\mathcal{M}_{F_j}$  par construction. Les hommes de  $\mathcal{M}_{F_j}$  (pour  $F_j \in \mathcal{F}$ ) effectuent leur unique rencontre dès leur date de disponibilité. La femme  $F_j$  est disponible aux créneaux de  $\llbracket 0, r_j^{(2)} - 1 \rrbracket$  pour effectuer les rencontres avec les hommes de  $\mathcal{M}_{F_j}$ .

4. **Si une solution existe pour  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  alors  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  admet une solution.**

Si une solution avec  $W_{\max} = 0$  existe pour le problème associé à  $\mathcal{F}$  et  $\mathcal{M} \cup \mathcal{M}^+$ , alors les rencontres entre les femmes et les hommes de  $\mathcal{M}^+$  ont lieu exactement à la date de disponibilité des hommes. Ces rencontres ont lieu par construction avant la date de disponibilité des femmes dans la première instance et tous ces créneaux sont occupés par une rencontre. Les rencontres entre les femmes et les hommes de  $\mathcal{M}$  ont donc lieu à partir des dates de disponibilité des femmes. Ce qui nous donne une solution pour l'instance de départ.

On en déduit que le cas plus général  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max} \leq k$  est  $\mathcal{NP}$ -complet et ainsi que le problème d'optimisation associé  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max}$  est  $\mathcal{NP}$ -difficile.  $\square$

**Corollaire 5.3.3.** *Les problèmes suivants sont  $\mathcal{NP}$ -difficiles :*

- $1\text{-}1MS|(U \cup V, E), r_j^{(2)}|W_{\max}$  ;
- $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}|W_{\max}$  et  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_j^{(2)}|W_{\max}$ .

*Démonstration.* 5.3.3

- Le problème  $1\text{-}1MS|(U \cup V, E), r_j^{(2)}|W_{\max}$  est équivalent à  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max}$ . Il suffit d'inverser le rôle des hommes et des femmes (cf. chapitre 3 section 3.2.3).
- On réduit les problèmes  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max}$  et  $1\text{-}1MS|(U \cup V, E), r_j^{(2)}|W_{\max}$  (nombres d'hommes et de femmes quelconques) aux problèmes  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_i^{(1)}|W_{\max}$  et  $1\text{-}1MS|(U \cup V, E), |U| = |V|, r_j^{(2)}|W_{\max}$  (nombre de femmes égal au nombre d'hommes) en ajoutant des participants fictifs qui n'effectuent aucune rencontre (cf. chapitre 3 section 3.2.2).

$\square$

## 5.4 Cas avec rencontres interdites et sans retards

Nous démontrons maintenant que les problèmes dans lesquels des rencontres sont interdites et sans retards des participants sont  $\mathcal{NP}$ -difficiles quelle que soit la proportion d'hommes et de femmes. Ainsi, nous obtenons que l'ensemble des cas dans lesquels des rencontres sont interdites sont  $\mathcal{NP}$ -difficiles.

**Proposition 5.4.1.** *Le problème  $1\text{-}1MS|(U \cup V, E), |U| = |V|W_{\max}$  est  $\mathcal{NP}$ -difficile.*

Afin de démontrer ce résultat, nous réduisons polynomialement  $1\text{-}1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  qui est  $\mathcal{NP}$ -complet à  $1\text{-}1MS|(U \cup V, E), |U| = |V|W_{\max} = 0$ . L'idée de la réduction est d'ajouter des femmes fictives qui rencontrent les hommes ayant des retards avant leur date d'arrivée afin de supprimer leur retard. Cela n'est pas suffisant pour que les participants aient tous un nombre de créneaux d'attente nul. On ajoute alors des hommes fictifs pour combler les attentes des femmes fictives.

*Démonstration.* 5.4.1

Nous allons effectuer une réduction polynomiale de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  vers  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$ . Nous illustrons cette réduction sur l'exemple dont les données sont les suivantes :

- L'ensemble des hommes est  $\mathcal{M} = \{M_0, M_1, M_2, M_3, M_4, M_5\}$ .
- L'ensemble des femmes est  $\mathcal{F} = \{F_0, F_1, F_2\}$ .
- Les dates d'arrivée des hommes sont :  $r_0^{(1)} = r_1^{(1)} = r_2^{(1)} = 0$  ;  $r_3^{(1)} = r_4^{(1)} = 3$  et  $r_5^{(1)} = 5$ .
- Les femmes n'ont pas de retard.  $\forall j \in \llbracket 0, n-1 \rrbracket, r_j^{(2)} = 0$ .
- La matrice des rencontres  $(p_{ij})_{(i,j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket}$  est présentée dans le tableau 5.1. De manière exceptionnelle et dans un souci de lisibilité, les rencontres interdites sont représentées par des cases blanches dans cette démonstrations.

		$F_0$	$F_1$	$F_2$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0
$M_0$	0	1	1	1
$M_1$	0	1	1	1
$M_2$	0	1	1	
$M_3$	3	1		
$M_4$	3	1	1	
$M_5$	5	1		

Tableau 5.1 – Matrice des rencontres de l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ .

1. Il est clair que  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0 \in \mathcal{NP}$  : une solution OUI est vérifiable en temps polynomial. Il suffit de vérifier que toutes les rencontres sont bien autorisées, que les rencontres de chaque participant sont consécutives et démarrent à la date zéro et que toutes les autres contraintes sont respectées.

2. **Construction d'une instance de  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$  à partir d'une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ .**

Soit une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ . Nous posons quelques restrictions justifiées sur cette instance.

- On rappelle que la liste  $(r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}$  est ordonnée dans l'ordre croissant. En particulier, on a  $r_{m-1}^{(1)} = \max_{i \in \llbracket 0, m-1 \rrbracket} r_i^{(1)}$ .
- On suppose également que  $r_{m-1}^{(1)} \neq 0$  (sinon toutes les dates d'arrivée des hommes sont nulles et l'instance appartient à  $1-1MS|(U \cup V, E)|W_{\max} = 0$ ).
- D'après les inégalités 5.1 de la démonstration du lemme 5.3.1 et le fait que  $\forall j \in \llbracket 0, n-1 \rrbracket, r_j^{(2)} = 0$ , on sait que si l'instance admet une solution, alors les dates d'arrivée des hommes sont bornées polynomialement :

$$\forall i \in \llbracket 0, m-1 \rrbracket, r_i^{(1)} < m. \quad (5.4)$$

- Finalement, on admet que le nombre d'hommes arrivant à l'heure est exactement le nombre de femmes  $m$ . Dans le cas contraire, l'instance n'a pas de solution sans attentes car chacune des  $m$  femmes (qui arrive à l'heure) doit rencontrer un homme à la date 0. On en déduit immédiatement que  $m \geq n$ .

On construit une instance de  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$  à partir de l'instance avec retards pour les hommes comme suit :

- (a) L'ensemble des hommes  $\mathcal{M}$  est conservé. Pour chaque entier  $e$  de  $\llbracket 1, r_{m-1}^{(1)} \rrbracket \setminus \{r_n^{(1)}, \dots, r_{m-1}^{(1)}\}$ , on ajoute à l'ensemble  $\mathcal{M}$  un homme fictif, auquel on affecte une date d'arrivée fictive égale à  $e$ . L'ensemble des hommes de la nouvelle instance est noté  $\mathcal{M}' = \{M'_i \mid i \in \llbracket 0, m' - 1 \rrbracket\}$ . On ordonne  $\mathcal{M}'$  dans l'ordre croissant des dates d'arrivée. Ces dates d'arrivée fictives sont notées  $r_i'^{(1)}$ .

Dans notre exemple,  $r_{m-1}^{(1)} = 5$  et l'ensemble  $\{r_n^{(1)}, \dots, r_{m-1}^{(1)}\} = \{3, 5\}$ , on ajoute donc trois hommes fictifs dont les dates d'arrivée fictives respectives sont 1, 2 et 4. L'ordre des hommes de  $\mathcal{M}'$  de cardinal  $m' = 9$  est indiqué dans le tableau 5.2. Les participants réels sont indiqués en gras dans ce tableau.

- (b) On conserve les femmes de  $\mathcal{F}$  et on ajoute  $m' - n$  femmes fictives notées  $\{F'_j \mid j \in \llbracket n, m' - 1 \rrbracket\}$ . L'ensemble des femmes de la nouvelle instance est noté  $\mathcal{F}' = \{F'_j \mid j \in \llbracket 0, m' - 1 \rrbracket\}$  où  $F'_j = F_j$  si  $j \in \llbracket 0, n - 1 \rrbracket$ .

Dans notre exemple, on ajoute donc six femmes (notées  $F'_3$  à  $F'_8$ ) de manière à obtenir autant d'hommes que de femmes.

- (c) On construit la matrice des rencontres  $(P_{ij})_{(i,j) \in \llbracket 0, m' - 1 \rrbracket^2}$ . Soit  $(i, j) \in \llbracket 0, m' - 1 \rrbracket^2$ .

$$P_{ij} = \begin{cases} p_{ij} & \text{si } M_i \in \mathcal{M}' \text{ et } j \in \mathcal{F}; \\ 1 & \text{si } M'_i \in \mathcal{M}' \text{ et } j \in \llbracket \max(i - r_i'^{(1)} + 1, n), i \rrbracket; \\ 0 & \text{sinon.} \end{cases}$$

Cette matrice est définie de manière à ce que chaque homme ait exactement autant de rencontres avec des femmes fictives qu'il a de créneaux de retard dans l'instance de départ. Dans notre exemple, la matrice des rencontres est celle présentée dans le tableau 5.2.

$r_i'^{(1)}$		$F'_0$	$F'_1$	$F'_2$	$F'_3$	$F'_4$	$F'_5$	$F'_6$	$F'_7$	$F'_8$
0	<b><math>M'_0</math></b>	1	1	1						
0	<b><math>M'_1</math></b>	1	1	1						
0	<b><math>M'_2</math></b>	1	1							
1	$M'_3$				1					
2	$M'_4$				1	1				
3	<b><math>M'_5</math></b>	1			1	1	1			
3	<b><math>M'_6</math></b>	1	1			1	1	1		
4	$M'_7$					1	1	1	1	
5	<b><math>M'_8</math></b>	1				1	1	1	1	1

Tableau 5.2 – Matrice des rencontres de l'instance de  $1-1MS|(U \cup V, E)$ ,  $|U| = |V|$ ,  $W_{\max} = 0$ .

- (d) Dans la nouvelle instance, les dates d'arrivée de tous les participants sont fixées à zéro.

Cette construction est bien polynomiale. Le nombre d'hommes fictifs que l'on ajoute à la soirée est égal à  $\text{card}(\llbracket 1, r_{m-1}^{(1)} \rrbracket \setminus \{r_i^{(1)} \mid i \in \llbracket 0, m - 1 \rrbracket\}) \leq r_{m-1}^{(1)} - 1 \leq m - 2$  car  $r_{m-1}^{(1)} < m$  d'après 5.4. Le nombre total d'hommes  $m'$  est donc inférieur ou égal à  $m + m - 2 = 2m - 2$ . Le nombre de femmes fictives est donc égal à  $m' - n \leq 2m - 2 - n$ . Le nombre de participants fictifs ajoutés est donc borné polynomialement par  $3m - n - 4$ .

3. **Si une solution existe pour  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  alors il existe une solution pour  $1-1MS|(U \cup V, E), |U| = |V|, W_{\max} = 0$ .**

S'il existe une solution pour l'instance de départ :  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ , alors on construit une solution pour l'instance de  $1-1MS|(U \cup V, E), |U| = |V|, W_{\max} = 0$  de la manière suivante. Les dates de rencontres entre deux participants réels sont conservées. On planifie la rencontre entre un homme  $M'_i$  ( $i \in \llbracket n, m' - 1 \rrbracket$ ) et une femme fictive  $F'_j$  ( $j \in \llbracket n, m' - 1 \rrbracket$ ) qui doit le rencontrer à la date  $(j - i)$ . Par construction, un homme réel  $M_i$  a exactement  $r_i^{(1)}$  rencontres à effectuer avant sa date d'arrivée et il n'a donc pas d'attentes. De même, les femmes n'ont pas d'attentes par construction.

Le tableau 5.3 présente une solution de l'instance de  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$  pour notre exemple construite à partir d'une solution de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  composée des trois premières colonnes.

	$F'_0$	$F'_1$	$F'_2$	$F'_3$	$F'_4$	$F'_5$	$F'_6$	$F'_7$	$F'_8$
$M'_0$	2	0	1						
$M'_1$	1	2	0						
$M'_2$	0	1							
$M'_3$				0					
$M'_4$				1	0				
$M'_5$	3			2	1	0			
$M'_6$	4	3			2	1	0		
$M'_7$					3	2	1	0	
$M'_8$	5				4	3	2	1	0

Tableau 5.3 – Solution pour le problème  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$ .

4. **Si une solution existe pour  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$  alors il existe une solution pour  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$ .**

Si une solution existe pour l'instance de  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$ , alors l'affectation des rencontres des femmes fictives est unique. En effet, la dernière femme  $F'_{m'-1}$  a une seule rencontre avec l'homme  $M'_{m'-1}$  qu'elle doit nécessairement rencontrer à la date 0 pour qu'elle n'ait pas d'attentes. L'avant dernière femme  $F'_{m'-2}$  a au plus deux rencontres avec  $M'_{m'-2}$  et  $M'_{m'-1}$ , ce dernier étant occupé à la date 0, la rencontre entre  $F'_{m'-2}$  et  $M'_{m'-2}$  a lieu à la date 0 et la rencontre entre  $F'_{m'-2}$  et  $M'_{m'-1}$  à la date 1 (si elle a lieu). En raisonnant de cette manière pour chacune des femmes fictives, on se rend compte que les rencontres de la diagonale principale de la matrice  $(P_{ij})_{n \leq i, j \leq m'-1}$  ont nécessairement lieu à la date 0. De la même manière, toutes les rencontres de la  $k$ -ème sous-diagonale ont lieu à la date  $k$ . On vérifie facilement que toutes ces rencontres ont lieu avant la date d'arrivée des hommes réels. On obtient une solution de l'instance de  $1-1MS|(U \cup V, E), r_i^{(1)}|W_{\max} = 0$  en supprimant les rencontres des femmes fictives.

Nous avons ainsi établi que  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} = 0$  est  $\mathcal{NP}$ -complet. On en déduit que le problème plus général  $1-1MS|(U \cup V, E), |U| = |V||W_{\max} \leq k$  est aussi  $\mathcal{NP}$ -complet et ainsi, le problème d'optimisation associé est  $\mathcal{NP}$ -difficile.  $\square$

**Corollaire 5.4.2.** *Le problème  $1-1MS|(U \cup V, E)|W_{\max}$  est  $\mathcal{NP}$ -difficile.*

*Démonstration.* 5.4.2

Ces résultats se déduisent de la proposition 5.4.1 et du graphe de réduction de la figure 3.1a.  $\square$

À l'issue de cette partie, nous avons établi l'ensemble des résultats du tableau 5.4.

	$\emptyset$	$r_i^{(1)}$	$r_j^{(2)}$	$r_i^{(1)}, r_j^{(2)}$
$(U \cup V, E),  U  =  V $	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile
$(U \cup V, E)$	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile

Tableau 5.4 – Problèmes de speed-dating  $\mathcal{NP}$ -difficiles.

Les deux sections suivantes précisent la difficulté des problèmes de speed-dating avec rencontres interdites.

## 5.5 $\mathcal{NP}$ -difficulté au sens fort

**Proposition 5.5.1.** *Les problèmes de speed-dating avec rencontres interdites sont  $\mathcal{NP}$ -difficiles au sens fort.*

*Démonstration. 5.5.1*

On rappelle quelques définitions tirées de (Garey et Johnson, 1979). La fonction “ $Length[I]$ ” représente la longueur de l’instance  $I$  et la fonction “ $Max[I]$ ” représente l’“ampleur” du plus grand nombre de  $I$ . Pour un problème  $\Pi$  et un polynôme  $p$ , on note  $\Pi_p$  le sous-problème de  $\Pi$  restreint aux instances  $I$  telles que  $Max[I] \leq p(Length[I])$ . Un problème  $\Pi$  est  $\mathcal{NP}$ -difficile au sens fort si et seulement si il existe un polynôme  $p$  tel que  $\Pi_p$  soit  $\mathcal{NP}$ -difficile.

On définit  $\Pi$  comme le problème  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max}$ . Une instance  $I$  de  $\Pi$  peut être représentée par une matrice de booléens de taille  $m \times n$  (la matrice des rencontres) et de deux listes d’entiers de tailles  $m$  et  $n$  (dates d’arrivée des participants). Choisissons un polynôme  $p$  tel que le problème sans retards  $1-1MS|(U \cup V, E)|W_{\max}$  soit inclus dans  $\Pi_p$ . Pour cela, on pose le polynôme  $p(x) = x$ . Si  $I \in 1-1MS|(U \cup V, E)|W_{\max}$ , alors  $Max[I] = 1$  (l’instance est composée d’une matrice de booléens et de deux listes de zéros). On a ainsi  $Length[I] = mn + m + n$  (taille de l’instance) et  $Max[I] = 1 \leq p(Length[I]) = mn + m + n$ . Le problème  $\Pi_p$  est  $\mathcal{NP}$ -difficile car il contient le problème  $1-1MS|(U \cup V, E)|W_{\max}$  (qui est  $\mathcal{NP}$ -difficile comme nous l’avons démontré dans la proposition 5.4.1). Ainsi, le cas général  $1-1MS|(U \cup V, E)|W_{\max}$  est  $\mathcal{NP}$ -difficile au sens fort. On démontre de la même manière que tous les sous cas avec rencontres interdites sont  $\mathcal{NP}$ -difficiles au sens fort.  $\square$

## 5.6 Non approximabilité

Parmi les problèmes d’optimisation  $\mathcal{NP}$ -difficiles, on distingue plusieurs catégories de problèmes. Dans certains cas, si on ne connaît pas d’algorithme polynomial pour résoudre le problème à l’optimum, il est quand même possible de trouver un algorithme qui fournit une solution en temps polynomial dont on contrôle l’objectif. Pour un problème de minimisation  $\Pi$ , on dit qu’un algorithme noté  $A$  est un *algorithme d’approximation* s’il existe un nombre  $\rho > 1$  appelé *facteur d’approximation* tel que quelle que soit  $I$  une instance de  $\Pi$ , on a  $Opt(I) \leq A(I) \leq \rho Opt(I)$  où  $A(I)$  est la valeur de l’objectif trouvée par  $A$  et  $Opt(I)$  est la valeur optimale de l’objectif de  $I$ . La classe APX est l’ensemble des problèmes d’optimisation qui admettent un algorithme d’approximation.

**Proposition 5.6.1.** *Si  $\mathcal{P} \neq \mathcal{NP}$ , les problèmes de speed-dating avec rencontres interdites n’appartiennent pas à APX.*

*Démonstration. 5.6.1*

Raisonnons par contradiction. Supposons qu’il existe un algorithme d’approximation polynomial  $A$  de facteur d’approximation  $\rho > 1$  pour un problème de speed-dating avec rencontres interdites. Pour une instance  $I$ , si  $Opt(I) = 0$ , alors par définition :  $A(I) \leq \rho Opt(I)$  et  $A(I) = 0$ . Si  $Opt(I) \neq 0$ , alors  $A(I) \geq Opt(I) > 0$  et donc  $A(I) \neq 0$ . L’algorithme  $A$  induit donc un algorithme qui décide en temps polynomial si une instance a pour objectif 0. Cela entre en contradiction avec la  $\mathcal{NP}$ -complétude de l’ensemble des problèmes de décision de non attente des participants associés aux problèmes de speed-dating avec rencontres interdites.  $\square$

# Conclusion

Dans cette partie, nous avons établi la complexité de différents problèmes de speed-dating. Nous avons notamment démontré la  $\mathcal{NP}$ -complétude de certains problèmes de décision associés. On en déduit que les problèmes d'optimisation correspondants sont  $\mathcal{NP}$ -difficiles. Le tableau 5.5 résume les résultats de complexité établis dans cette partie.

	$\emptyset$	$r_i^{(1)}$	$r_j^{(2)}$	$r_i^{(1)}, r_j^{(2)}$
$K_{m,m}$	$O(m^2)$	$O(m^2)$	$O(m^2)$	OUVERT
$K_{m,n}$	$O(mn)$	$O(m^{5/2}n^{3/2}\log(mn))$	$O(n^{5/2}m^{3/2}\log(mn))$	OUVERT
$(U \cup V, E),  U = V $	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile
$(U \cup V, E)$	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile	$\mathcal{NP}$ -difficile

Tableau 5.5 – Résultats de complexité de problèmes de speed-dating pour l'objectif  $W_{\max}$ .

À la vue de ces résultats, il apparaît clairement que les rencontres interdites rendent le problème difficile. Ce qui rend faciles les problèmes sans rencontres interdites avec retards pour une population, c'est que les dates d'échéance de la population arrivant à l'heure est une constante et que les dates d'échéance des participants arrivant en retard ne correspondent qu'à une translation des dates d'arrivée de ceux-ci. Les techniques utilisées pour établir nos différents résultats de complexité ne sont pas applicables au cas ouvert sans rencontres interdites et avec retards possibles pour tous les participants. En effet, d'une part il n'existe pas de problèmes, d'open shop par exemple, dans la littérature considérant des retards pour les machines. D'autre part, l'ensemble des réductions de  $\mathcal{NP}$ -complétude exploitent largement la possibilité d'ajouter des participants ayant des rencontres interdites.

Pour clore la complexité des problèmes de speed-dating, nous envisageons dans un premier temps, l'étude du problème de décision  $1\text{-}1MS|K_{m,n}, r_i^{(1)}, r_j^{(2)}|W_{\max} = 0$  qui a des propriétés intéressantes (par exemple, si  $W_{\max} = 0$ , la liste des dates d'arrivée des femmes est incluse dans la liste des dates d'arrivée des hommes si  $m \leq n$ ). La résolution de ce cas peut apporter des pistes de recherche sur le cas général d'optimisation.

*Les résultats de complexité démontrés dans cette partie ont fait l'objet d'une présentation lors de la conférence ROADEF'12 (Le Roux et al., 2012), et ils font l'objet d'un article accepté dans la revue internationale Journal of the Operational Research Society.*



## **Troisième partie**

# **Bornes inférieures et bornes supérieures**





# Chapitre 6

## Bornes inférieures

### Sommaire

<b>6.1</b>	<b>Introduction</b>	<b>57</b>
<b>6.2</b>	<b>Borne inférieure dépendant des dates d'échéance des rencontres</b>	<b>58</b>
<b>6.3</b>	<b>Borne inférieure dépendant des dates d'arrivée des participants</b>	<b>59</b>
<b>6.4</b>	<b>Borne inférieure issue de couplages maximums</b>	<b>62</b>
<b>6.5</b>	<b>Condition nécessaire de non attente des participants</b>	<b>64</b>
<b>6.6</b>	<b>Résultats expérimentaux</b>	<b>65</b>
<b>6.7</b>	<b>Conclusion</b>	<b>66</b>

### 6.1 Introduction

Comme nous l'avons établi dans le chapitre précédent, les problèmes de speed-dating que nous considérons avec rencontres interdites sont  $\mathcal{NP}$ -difficiles. De plus il n'existe pas d'algorithme d'approximation en temps polynomial pour les résoudre (sauf si  $\mathcal{P} = \mathcal{NP}$ ). Il peut donc être intéressant de pouvoir fournir une évaluation de la valeur de l'objectif rapidement. Dans ce chapitre nous abordons la recherche de bornes inférieures pour le nombre maximal de créneaux d'attente des participants. En effet, dans toute démarche de minimisation d'un objectif, il est toujours pertinent d'établir une évaluation par défaut de la meilleure solution. Du point de vue du décideur, cela peut avoir un impact fort sur la première estimation attendue et du point de vue de la résolution, ces bornes inférieures permettront d'évaluer directement la qualité des solutions. Elles seront de plus des éléments forts de convergence pour les méthodes présentées par la suite.

De manière générale, la recherche d'une borne inférieure s'appuie sur la relaxation d'une partie du problème dans le but de diminuer la complexité de celui-ci. Cependant, la relaxation du problème engendre de la perte d'information et l'une des principales difficultés est de trouver le bon compromis entre la difficulté du sous-problème et la qualité de la borne inférieure qu'il détermine. Le problème relâché obtenu est ensuite traité dans son intégralité ou sous la forme de sous-problèmes. Dans ce chapitre nous allons nous intéresser à la relaxation de différentes contraintes des problèmes de speed-dating et étudier les sous-problèmes qui en découlent afin de déterminer des bornes inférieures rapides à calculer.

De nombreuses approches de recherche de bornes inférieures ont été traitées dans la littérature. Nous nous intéressons ici à celles qui ont un lien pertinent avec notre problème.

Parmi les contraintes les plus relâchées dans les problèmes d'ordonnancement, on trouve les contraintes de disjonction puisqu'elles apparaissent dans tous les problèmes d'ordonnancement d'atelier. On peut déterminer une borne triviale du  $C_{\max}$  pour le problème d'open shop  $O|pmtn|C_{\max}$  en relâchant les contraintes de disjonction des jobs ou des machines (Brucker, 2001). La relaxation des contraintes de disjonction des ressources est également utilisée pour le problème  $\mathcal{NP}$ -difficile de flow shop  $F2||\sum C_i$  (Brucker, 2001). De manière plus fine, on peut relâcher une partie des contraintes de disjonction des jobs ou des machines. On peut notamment citer l'article (Guéret et Prins,

1999) qui traite du problème d'open shop  $O||C_{\max}$  en relâchant les contraintes de disjonction de tous les jobs à l'exception d'un seul, ou (Gharbi et Mahjoubi, 2013) dans lequel les auteurs proposent de relâcher des contraintes de disjonction de toutes les machines sauf deux pour un problème de flow shop à permutation minimisant le  $C_{\max}$ . Trois des bornes que nous allons présenter sont basées sur la relaxation des contraintes de disjonction des participants.

Dans les sections 6.2, 6.3 et 6.4, on relâche les contraintes de disjonction des participants à plusieurs niveaux et on exploite les propriétés du problème pour déterminer des bornes inférieures de l'objectif  $W_{\max}$ . Dans la section 6.5, on s'appuie sur une approche dans laquelle on relâche la contrainte qui impose que chaque rencontre n'a lieu qu'une fois dans la soirée. Enfin, dans la section 6.6, nous comparons les bornes inférieures calculées sur notre ensemble d'instances.

## 6.2 Borne inférieure dépendant des dates d'échéance des rencontres $LB_0$

La première borne inférieure que nous présentons se base sur la relaxation des contraintes disjonctives, c'est-à-dire qu'on relâche la contrainte imposant que chaque participant ne peut pas effectuer plus d'une rencontre à la même date. Chaque rencontre a une liste de dates à laquelle elle peut être affectée sans provoquer d'attente pour les participants. Dans certains cas, la modélisation induite par cette relaxation permet de détecter que cette liste est vide et de déduire une borne inférieure pour le nombre d'attentes des deux participants.

La proposition 6.2.2 spécifie cette borne inférieure. Elle découle de la proposition 6.2.1 que nous énonçons maintenant et qui procure une écriture de l'objectif en fonction des dates de disponibilité et d'échéance des participants.

On rappelle quelques notations présentées dans la section 1.3. La valeur  $d_{ij}$  est la date d'échéance de la rencontre entre l'homme  $M_i \in \mathcal{M}$  et la femme  $F_j \in \mathcal{F}$ , il s'agit du minimum entre les dates de fin au plus tôt des deux participants. La valeur  $d_{\min}$  est la plus petite date d'échéance des rencontres. On note  $\mathcal{M}_{F_j}$  l'ensemble des hommes que rencontre  $F_j$  et  $\mathcal{F}_{M_i}$  l'ensemble des femmes que rencontre l'homme  $M_i$ . On rappelle que l'ensemble des hommes et l'ensemble des femmes sont classés par ordre de dates d'arrivée croissantes.

### Proposition 6.2.1.

$$W_{\max} = \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}. \quad (6.1)$$

*Démonstration.* 6.2.1

Rappelons que la définition de l'objectif est la suivante :

$$W_{\max} = \max(\max_{i \in \mathcal{M}} \{C_i^{(1)} - e_i^{(1)}\}, \max_{j \in \llbracket 0, n-1 \rrbracket} \{C_j^{(2)} - e_j^{(2)}\}).$$

On pose  $W := \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$ . Démontrons que  $W = W_{\max}$ .

#### 1. $W \leq W_{\max}$ .

Soit un couple  $(i_0, j_0) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket$  tel que  $p_{i_0 j_0} = 1$ . Si  $e_{j_0}^{(2)} \leq e_{i_0}^{(1)}$ , alors  $C_{i_0 j_0} - d_{i_0 j_0} = C_{i_0 j_0} - e_{j_0}^{(2)}$ , sinon  $C_{i_0 j_0} - d_{i_0 j_0} = C_{i_0 j_0} - e_{i_0}^{(1)}$ . Dans le premier cas,

$$C_{i_0 j_0} - d_{i_0 j_0} \leq \max\{C_{ij_0} \mid i \in \llbracket 0, m-1 \rrbracket, p_{ij_0} = 1\} - e_{j_0}^{(2)} = W_{j_0}^2 \leq W_{\max}.$$

Dans le deuxième cas,

$$C_{i_0 j_0} - d_{i_0 j_0} \leq \max\{C_{i_0 j} \mid j \in \llbracket 0, n-1 \rrbracket, p_{i_0 j} = 1\} - e_{i_0}^{(1)} = W_{i_0}^1 \leq W_{\max}.$$

Donc  $W = \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\} \leq W_{\max}$ .

#### 2. $W_{\max} \leq W$ .

Par définition,  $W_{\max} = \max\{W_i^{(1)}, W_j^{(2)} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket\}$ . Supposons que le maximum est atteint par la femme  $F_{j_0} \in \mathcal{F}$ . On a alors :

$$W_{\max} = C_{j_0}^2 - e_{j_0}^{(2)} = \max\{C_{ij_0} - e_{j_0}^{(2)} \mid i \in \llbracket 0, m-1 \rrbracket, p_{ij_0} = 1\}.$$

De plus, quel que soit  $i \in \mathcal{M}$  tel que  $p_{ij_0} = 1$ , on a  $e_{j_0}^{(2)} \geq \min(e_i^{(1)}, e_{j_0}^{(2)}) = d_{ij_0}$ , donc

$$\forall i \in \llbracket 0, m-1 \rrbracket, C_{ij_0} - e_{j_0}^{(2)} \leq C_{ij_0} - d_{ij_0} \leq \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}.$$

Donc  $W_{\max} = \max\{C_{ij_0} - e_{j_0}^{(2)} \mid i \in \llbracket 0, m-1 \rrbracket, p_{ij_0} = 1\} \leq W$ .

De même, on démontre que  $W_{\max} \leq W$  si  $W_{\max} = W_{\max}^{(1)}$ .

□

On en déduit la borne inférieure suivante :

**Proposition-Définition 6.2.2.** *La valeur  $\max\{0, r_{ij} - d_{ij} + 1 \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$  est une borne inférieure de l'objectif  $W_{\max}$ . On note cette borne inférieure  $LB_0$ .*

*Démonstration.* 6.2.2

Ce résultat se déduit directement de l'écriture de l'objectif  $W_{\max}$  donnée dans la proposition 6.2.1. D'après celle-ci, on a :

$$W_{\max} = \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}.$$

Or,  $\forall i \in \llbracket 0, m-1 \rrbracket, \forall j \in \llbracket 0, n-1 \rrbracket$ , tels que  $p_{ij} = 1$ , on a  $C_{ij} \geq r_{ij} + 1$ . On en déduit facilement que :

$$W_{\max} \geq \max\{r_{ij} - d_{ij} + 1 \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}.$$

D'où le résultat.

□

**Exemple 6.2.1.** On considère l'instance suivante, à trois hommes et quatre femmes pour laquelle les dates d'arrivée des hommes sont respectivement 0, 1 et 2 et les dates d'arrivée des femmes sont respectivement 0, 1, 3 et 4. La rencontre entre  $M_1$  et  $F_2$  est interdite. La figure 6.1 représente une solution optimale de cette instance.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	1	3	4	$W_i^{(1)}$
$M_0$	0	0	2	3	4	1
$M_1$	1	2	1		5	2
$M_2$	2	4	3	5	6	1
	$W_j^{(2)}$	2	0	1	0	$W_{\max}^* = 2$

FIGURE 6.1 – Solution optimale avec  $W_{\max}^* = 2$ .

Pour cette instance,  $LB_0 = 1$ . En effet,  $r_{0,3} = \max(r_0^{(1)}, r_3^{(2)}) = \max(0, 4) = 4$  et  $d_{0,3} = \min(e_0^{(1)}, e_3^{(2)}) = \min(4, 7) = 4$ , donc  $LB_0 \geq r_{0,3} - d_{0,3} + 1 = 1$ . De même, on trouve,  $r_{1,3} - d_{1,3} + 1 = 1$ . On trouve pour le reste des rencontres :  $r_{ij} - d_{ij} + 1 \leq 0$ .

### 6.3 Borne inférieure dépendant des dates d'arrivée des participants $LB_1$

Dans la section précédente, nous avons relâché l'ensemble des contraintes disjonctives qui sont des contraintes fortes du problème. Cette relaxation entraîne la perte d'une part importante de la structure du problème. Dans cette section, nous affinons la relaxation afin d'obtenir une borne inférieure de meilleure qualité. L'idée est de relâcher les contraintes disjonctives sur une des populations et de résoudre le problème de minimisation des attentes de l'autre population. Si on autorise chacune des femmes à effectuer plusieurs rencontres à la même date, les planifications des rencontres de chacun des hommes deviennent indépendantes. Dans ce cas, la planification des rencontres de l'un des hommes est équivalente au problème classique d'ordonnancement à une machine  $1|r_i, p_i = 1|C_{\max}$ . En effet, les jobs représentent les rencontres de l'homme et les dates de disponibilité des jobs représentent les dates de disponibilité des rencontres. Minimiser le nombre de créneaux d'attente de l'homme considéré est équivalent dans ce cas à minimiser la date de fin de ses rencontres. Le problème  $1|r_i, p_i = 1|C_{\max}$  se résout trivialement en planifiant chaque job par ordre croissant de date de disponibilité (Brucker, 2001). On procède de même en relâchant les contraintes de disjonction des

hommes pour traiter le sous-problème lié à chaque femme. On a ainsi  $m + n$  problèmes  $1|r_i, p_i = 1|C_{\max}$  à résoudre.

**Proposition 6.3.1.** *Soit un homme  $M_i \in \mathcal{M}$ . Une borne inférieure de son nombre d'attentes est déterminée par la valeur :  $\max\{r_{\psi_i(j)}^{(2)} - j - r_i^{(1)} \mid j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket; p_{i, \psi_i(j)} = 1\}$  où  $F_{\psi_i(j)}$  est la  $j$ -ème femme que rencontre l'homme  $M_i$  pour  $0 \leq j \leq |\mathcal{F}_{M_i}| - 1$ .*

*Démonstration.* 6.3.1 Minimiser le nombre d'attentes de  $M_i$  est équivalent à la résolution du problème d'ordonnement  $1|r_i, p_i = 1|C_{\max}$  dans lequel l'homme représente la machine, les jobs représentent les rencontres de cet homme et les dates de disponibilité des jobs représentent les dates de disponibilité des rencontres ( $r_{ij}$ ). Trivialement, minimiser la date de fin des rencontres de  $M_i$  (ou  $C_{\max}$ ) implique nécessairement la minimisation de ses attentes car il n'y a qu'un homme. Calculons la date de fin de ses rencontres en planifiant ses rencontres dans l'ordre d'arrivée des femmes. On a :

$$C_{i, \psi_i(j)} = \begin{cases} \max(r_i^{(1)}, r_{\psi_i(0)}^{(2)}) + 1 & \text{si } j = 0, \\ \max(1 + C_{i, \psi_i(j-1)}, 1 + r_{\psi_i(j)}^{(2)}) & \text{si } 1 \leq j \leq |\mathcal{F}_{M_i}| - 1. \end{cases}$$

L'homme  $M_i$  rencontre la femme  $F_{\psi_i(j)}$  juste après sa rencontre avec la femme  $F_{\psi_i(j-1)}$  si  $F_{\psi_i(j)}$  est disponible, sinon il attend que cette dernière soit arrivée. On démontre par récurrence que :

$$\forall j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket, \quad C_{i, \psi_i(j)} = \max(r_i^{(1)} + j + 1, \max_{k \in \llbracket 0, j \rrbracket} \{j + 1 - k + r_{\psi_i(k)}^{(2)}\}).$$

La dernière rencontre qu'effectue l'homme  $M_i$  est avec la femme  $F_{\psi_i(|\mathcal{F}_{M_i}|-1)}$  donc il finit ses rencontres à la date :

$$C_i^{(1)} = C_{i, \psi_i(|\mathcal{F}_{M_i}|-1)} = \max(r_i^{(1)} + |\mathcal{F}_{M_i}|, \max_{j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket} \{|\mathcal{F}_{M_i}| - j + r_{\psi_i(j)}^{(2)}\}).$$

On en déduit que le nombre d'attentes de l'homme  $M_i$  est :

$$\begin{aligned} W_i^{(1)} &= C_{i, \psi_i(|\mathcal{F}_{M_i}|-1)} - e_i^{(1)} \\ &= \max(r_i^{(1)} + |\mathcal{F}_{M_i}|, \max_{j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket} \{|\mathcal{F}_{M_i}| - j + r_{\psi_i(j)}^{(2)}\}) - (r_i^{(1)} + |\mathcal{F}_{M_i}|) \\ &= \max(0, \max_{j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket} \{r_{\psi_i(j)}^{(2)} - j\} - r_i^{(1)}). \end{aligned}$$

D'où le résultat. □

**Proposition-Définition 6.3.2.** *Une borne inférieure pour le nombre maximal des attentes des participants est :  $\max(0, \max\{r_{\psi_i(j)}^{(2)} - j - r_i^{(1)} \mid i \in \llbracket 0, m - 1 \rrbracket, j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket, p_{i, \psi_i(j)} = 1\}, \max\{r_{\varphi_j(i)}^{(1)} - i - r_j^{(2)} \mid j \in \llbracket 0, n - 1 \rrbracket, i \in \llbracket 0, |\mathcal{M}_{F_j}| - 1 \rrbracket, p_{\varphi_j(i), j} = 1\})$  où  $F_{\psi_i(j)}$  est la  $j$ -ème femme que rencontre l'homme  $M_i$  pour  $0 \leq j \leq |\mathcal{F}_{M_i}| - 1$  et  $M_{\varphi_j(i)}$  est le  $i$ -ème homme que rencontre la femme  $F_j$  pour  $0 \leq i \leq |\mathcal{M}_{F_j}| - 1$ . Cette borne inférieure est notée  $LB_1$ .*

*Démonstration.* 6.3.2 D'après la proposition 6.3.1, on connaît une borne inférieure du nombre d'attentes de chacun des hommes qui est :

$$\max\{r_{\psi_i(j)}^{(2)} - j - r_i^{(1)} \mid j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket; p_{i, \psi_i(j)} = 1\}$$

où  $F_{\psi_i(j)}$  est la  $j$ -ème femme que rencontre l'homme  $M_i$  pour  $0 \leq j \leq |\mathcal{F}_{M_i}| - 1$ . De la même manière, on démontre qu'une borne inférieure du nombre d'attentes d'une femme  $F_j \in \mathcal{F}$  est

$$\max\{r_{\varphi_j(i)}^{(1)} - i - r_j^{(2)} \mid i \in \llbracket 0, |\mathcal{M}_{F_j}| - 1 \rrbracket; p_{\varphi_j(i), j} = 1\}$$

où  $M_{\varphi_j(i)}$  est le  $i$ -ème homme que rencontre la femme  $F_j$  pour  $0 \leq i \leq |\mathcal{M}_{F_j}| - 1$ . En agrégeant ces bornes, on obtient le résultat. □

**Exemple 6.3.1.** La figure 6.2 présente une instance à trois hommes, deux femmes avec deux rencontres interdites et une solution optimale pour cette instance.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	2	2	3	$W_i^{(1)}$
$M_0$	0	0	2	5	3	2
$M_1$	0		3	2	4	2
$M_2$	1	1	4		5	2
	$W_j^{(2)}$	0	0	2	0	$W_{\max}^* = 2$

FIGURE 6.2 – Solution optimale avec  $W_{\max}^* = 2$ .

La figure 6.3 présente un ordonnancement optimal pour chacun des hommes dans lequel on a relâché les contraintes de non ubiquité des femmes. On remarque ici que l'homme  $M_2$  a deux créneaux d'attentes, une borne inférieure de l'objectif est donc 2.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	2	2	3	$W_i^{(1)}$
$M_0$	0	0	2	3	4	1
$M_1$	0		2	3	4	2
$M_2$	1	1	2		3	0
	$W_j^{(2)}$					$LB_1 \geq 2$

FIGURE 6.3 – Relaxation des contraintes disjonctives pour les femmes

Dans la figure 6.4, les contraintes de non ubiquité des hommes sont relâchées. Ainsi, chacune des femmes a un planning optimal. Dans ce cas, aucune des femmes n'a de créneau d'attentes. On en déduit que  $LB_1 = 2$ . Notons que  $LB_0 = 1$  pour cette instance ce qui illustre la dominance de  $LB_1$  sur  $LB_0$ , établie ci-après dans la proposition 6.3.4.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	2	2	3	$W_i^{(1)}$
$M_0$	0	0	2	2	3	
$M_1$	0		3	3	4	
$M_2$	1	1	4		5	
	$W_j^{(2)}$	0	0	0	0	$LB_1 \geq 0$

FIGURE 6.4 – Relaxation des contraintes disjonctives pour les hommes

D'après la proposition-définition 6.3.2,  $LB_1$  est le maximum d'un ensemble de cardinal en  $O(mn)$ . En effet, chacun des ensembles  $\{r_{\psi_i(j)}^{(2)} - j - r_i^{(1)} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, |\mathcal{F}_{M_i}| - 1 \rrbracket, p_{i, \psi_i(j)} = 1\}$  et  $\{r_{\varphi_j(i)}^{(1)} - i - r_j^{(2)} \mid j \in \llbracket 0, n-1 \rrbracket, i \in \llbracket 0, |\mathcal{M}_{F_j}| - 1 \rrbracket, p_{\varphi_j(i), j} = 1\}$  contient au plus  $mn$  éléments. Dans le cas où il n'y a pas de rencontres interdites, on peut grandement simplifier ces calculs en passant à un ensemble de cardinal en  $O(m+n)$ , ce que nous énonçons dans la proposition suivante.

**Proposition 6.3.3.** *Dans le cas où aucune rencontre n'est interdite,  $LB_1 = \max(0, \max_{M_i \in \mathcal{M}} \{r_i^{(1)} - i\}, \max_{F_j \in \mathcal{F}} \{r_j^{(2)} - j\})$ .*

*Démonstration.* 6.3.3 Si on transcrit  $LB_1$  obtenue dans la proposition 6.3.2 dans le cas où aucune rencontre n'est interdite, on obtient :

$$LB_1 = \max(0, \max\{r_j^{(2)} - j - r_i^{(1)} \mid (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket\}, \max\{r_i^{(1)} - i - r_j^{(2)} \mid (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket\}).$$

Cependant, on voit immédiatement que :

$$\begin{aligned} \forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, \quad & r_j^{(2)} - j - r_i^{(1)} \leq r_j^{(2)} - j, \\ \text{et } \forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, \quad & r_i^{(1)} - i - r_j^{(2)} \leq r_i^{(1)} - i. \end{aligned}$$

On a donc :

$$\max\{r_j^{(2)} - j - r_i^{(1)} \mid (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket\} = \max\{r_j^{(2)} - j \mid j \in \llbracket 0, n-1 \rrbracket\}, \quad (6.2)$$

$$\text{et } \max\{r_i^{(1)} - i - r_j^{(2)} \mid (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket\} = \max\{r_i^{(1)} - i \mid i \in \llbracket 0, m-1 \rrbracket\}. \quad (6.3)$$

D'où le résultat.  $\square$

**Exemple 6.3.2.** On considère l'instance à trois hommes et quatre femmes sans rencontres interdites dont les dates d'arrivée sont les suivantes :  $r_0^{(1)} = r_1^{(1)} = 0$  ;  $r_2^{(1)} = 1$  ;  $r_0^{(2)} = 0$  ;  $r_1^{(2)} = r_2^{(2)} = 2$  et  $r_3^{(2)} = 3$ . La figure 6.5 représente une solution optimale pour cette instance.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	2	2	3	$W_i^{(1)}$
$M_0$	0	0	4	2	3	1
$M_1$	0	1	2	3	4	1
$M_2$	1	2	3	4	5	1
	$W_j^{(2)}$	0	0	0	0	$W_{\max}^* = 1$

FIGURE 6.5 – Solution optimale avec  $W_{\max}^* = 1$ .

La borne  $LB_2$  de cette instance est 1, en effet  $r_1^{(2)} - 1 = 1$ . Autrement dit, on ne peut pas planifier les rencontres de l'homme  $M_0$  sans attente car il n'y a que la femme  $F_0$  qui soit présente lors des créneaux 0 et 1. La borne précédemment présentée  $LB_0$  est nulle pour cette instance.

**Proposition 6.3.4.** Dans le cas général, la borne  $LB_1$  domine toujours la borne  $LB_0$ .

*Démonstration.* 6.3.4 Soit une instance avec rencontres interdites,

$$\begin{aligned} LB_0 &= \max\{0, r_{ij} - e_{ij} + 1 \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\} \\ &= \max\{0, \max(r_i^{(1)}, r_j^{(2)}) - \min(r_i^{(1)} + |\mathcal{F}_{M_i}|, r_j^{(2)} + |\mathcal{M}_{F_j}|) + 1 \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}. \end{aligned}$$

Soit un couple  $(i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket$  tel que  $p_{ij} = 1$ ,

$$\max(r_i^{(1)}, r_j^{(2)}) - \min(r_i^{(1)} + |\mathcal{F}_{M_i}|, r_j^{(2)} + |\mathcal{M}_{F_j}|) + 1 =$$

$$\begin{cases} r_j^{(2)} - r_j^{(2)} - |\mathcal{M}_{F_j}| + 1 = -|\mathcal{M}_{F_j}| + 1 \leq 0 & \text{si } r_j^{(2)} \geq r_i^{(1)} \text{ et } r_j^{(2)} + |\mathcal{M}_{F_j}| \leq r_i^{(1)} + |\mathcal{F}_{M_i}|, \\ r_i^{(1)} - r_i^{(1)} - |\mathcal{F}_{M_i}| + 1 = -|\mathcal{F}_{M_i}| + 1 \leq 0 & \text{si } r_j^{(2)} \leq r_i^{(1)} \text{ et } r_j^{(2)} + |\mathcal{M}_{F_j}| \geq r_i^{(1)} + |\mathcal{F}_{M_i}|, \\ r_j^{(2)} - r_i^{(1)} - |\mathcal{F}_{M_i}| + 1 & \text{si } r_j^{(2)} \geq r_i^{(1)} \text{ et } r_j^{(2)} + |\mathcal{M}_{F_j}| \geq r_i^{(1)} + |\mathcal{F}_{M_i}|, \\ r_i^{(1)} - r_j^{(2)} - |\mathcal{M}_{F_j}| + 1 & \text{si } r_j^{(2)} \leq r_i^{(1)} \text{ et } r_j^{(2)} + |\mathcal{M}_{F_j}| \leq r_i^{(1)} + |\mathcal{F}_{M_i}|. \end{cases}$$

Dans les deux premiers cas,  $LB_0 = 0 \leq LB_1$ . Dans les deux derniers cas,

$$\begin{aligned} r_j^{(2)} - r_i^{(1)} - |\mathcal{F}_{M_i}| + 1 &\leq r_j^{(2)} - \psi_i^{-1}(j) - r_i^{(1)} \leq LB_1 \\ \text{et } r_i^{(1)} - r_j^{(2)} - |\mathcal{M}_{F_j}| + 1 &\leq r_i^{(1)} - \varphi_j^{-1}(i) - r_j^{(2)} \leq LB_1. \end{aligned}$$

D'où le résultat.  $\square$

## 6.4 Borne inférieure issue de couplages maximums $LB_2$

Dans la section précédente, on a relâché les contraintes de disjonction de chacune des populations pour exploiter le fait que les sous-problèmes obtenus sont polynomiaux et obtenir une borne inférieure. Dans cette section, on opère les mêmes relaxations sur les contraintes disjonctives, mais avec un point de vue différent sur les sous-problèmes obtenus.

L'idée est d'exploiter les intervalles de dates disponibles pour chaque rencontre  $\llbracket r_{ij}, d_{ij} + LB_0 - 1 \rrbracket$ , intervalles que nous avons présentés dans la section 6.2 (on ajoute  $LB_0$  afin d'avoir des intervalles non vides). Au lieu de les exploiter individuellement comme dans le cas de  $LB_0$ , on étudie toutes les rencontres d'un participant en même temps.

Nous introduisons dans un premier temps une borne inférieure du nombre maximal d'attentes pour chaque participant individuellement.

**Proposition-Définition 6.4.1.** *Pour chaque homme  $M_i \in \mathcal{M}$  (resp. femme  $F_j \in \mathcal{F}$ ), on construit un graphe biparti  $G_{M_i}$  (resp.  $G_{F_j}$ ) dans lequel les sommets représentent les femmes que doit rencontrer  $M_i$  (resp. les hommes que doit rencontrer  $F_j$ ) et les dates de  $\llbracket r_i^{(1)}, e_i^{(1)} + LB_0 - 1 \rrbracket$  (resp.  $\llbracket r_j^{(2)}, e_j^{(2)} + LB_0 - 1 \rrbracket$ ). Une arête relie une femme  $F_j$  (resp. un homme  $M_i$ ) à chacun des sommets de  $\llbracket r_{ij}, d_{ij} + LB_0 - 1 \rrbracket$ . Pour chacun de ces graphes, on détermine le cardinal d'un couplage maximum. Ce cardinal est noté  $\text{card}_{M_i}$  (resp.  $\text{card}_{F_j}$ ). On pose  $\Delta_{M_i} := |\mathcal{F}_{M_i}| - \text{card}_{M_i}$  (resp.  $\Delta_{F_j} := |\mathcal{M}_{F_j}| - \text{card}_{F_j}$ ). La valeur  $\max\{\Delta_{M_i}, \Delta_{F_j} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket\} + LB_0$  est une borne inférieure de l'objectif  $W_{\max}$ . Cette borne inférieure est notée  $LB_2$ .*

*Démonstration.* 6.4.1 Soit  $M_i \in \mathcal{M}$ , démontrons que  $\Delta_{M_i} + LB_0$  est une borne inférieure de  $W_{\max}$ . Supposons que  $\Delta_{M_i} > 0$ . Cela signifie que  $M_i$  ne peut pas effectuer toutes ses rencontres en respectant les intervalles de date admissibles  $\llbracket r_{ij}, d_{ij} - 1 + LB_0 \rrbracket$ . Pour obtenir un couplage de cardinal égal au nombre de femmes que doit rencontrer  $M_i$ , il est nécessaire d'ajouter  $\Delta_{M_i}$  à toutes les bornes supérieures des intervalles de dates des rencontres de  $M_i$  et donc  $\Delta_{M_i} + LB_0$  est une borne inférieure de  $W_{\max}$ . De même, on démontre que pour toute femme  $F_j \in \mathcal{F}$ ,  $\Delta_{F_j} + LB_0$  est une borne inférieure de  $W_{\max}$ .  $\square$

La borne inférieure  $LB_2$  se calcule en  $O(mn)$ . En effet, les graphes  $G_{M_i}$  et  $G_{F_j}$  (pour  $M_i \in \mathcal{M}$  et  $F_j \in \mathcal{F}$ ) sont bipartis et convexes car l'ensemble des dates reliant un(e) participant(e) est un intervalle. On rappelle qu'un graphe biparti  $G = (U \cup V, E)$  est dit convexe si on peut ordonner les sommets de  $V$  de manière à ce que pour chaque sommet  $u$  de  $U$ , l'ensemble des sommets adjacents à  $u$  forment un intervalle. Le problème de couplage de cardinal maximal dans un graphe biparti convexe est linéaire en le nombre de sommets du graphes (Lipski et Preparata, 1981; Steiner et Yeomans, 1996). Dans le calcul de  $LB_2$ , on traite  $m$  graphes bipartis convexes de taille  $O(n)$  et  $n$  graphes bipartis convexes de taille  $O(m)$ . Ainsi le temps de calcul de  $LB_2$  est en  $O(mn)$ .

**Exemple 6.4.1.** On considère l'exemple à quatre hommes et cinq femmes dont les dates d'arrivée sont les suivantes :  $r_0^{(1)} = 0; r_1^{(1)} = 3; r_2^{(1)} = r_3^{(1)} = 4$  et  $r_0^{(2)} = r_1^{(2)} = 0; r_2^{(2)} = r_3^{(2)} = 1; r_4^{(2)} = 2$  et avec une rencontre interdite entre l'homme  $M_0$  et la femme  $F_2$ . On a  $LB_0 = 1$ . En effet,  $r_{2,0} = \max(r_2^{(1)}, r_0^{(2)}) = 4$  et  $e_{2,0} = \min(e_2^{(1)}, e_0^{(2)}) = 4$ . On notera également que  $LB_1 = 2$ . En effet, on ne peut pas ordonnancer les rendez-vous de l'homme  $M_2$  sans qu'un participant ait au moins deux créneaux d'attente.

La figure 6.6 représente les dates disponibles pour chaque rencontre si  $W_{\max} \leq LB_0 = 1$ . Chaque case  $(M_i, F_j)$  contient l'intervalle  $\llbracket r_{ij}, e_{ij} + LB_0 - 1 \rrbracket$  qui représente l'ensemble des dates auxquelles peut avoir lieu cette rencontre si  $W_{\max} \leq LB_0$ . L'emploi de la borne inférieure  $LB_0$  permet d'obtenir pour chaque rencontre un intervalle non vide de dates de rencontres. En effet, dans cet exemple,  $\llbracket r_{2,0}, e_{2,0} - 1 \rrbracket = \llbracket 4, 3 \rrbracket := \emptyset$ .


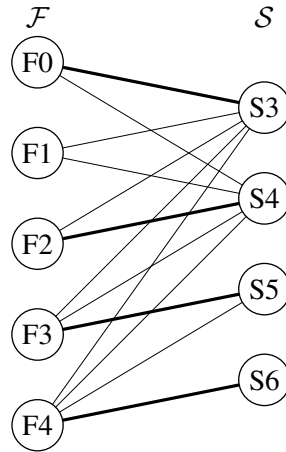
		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1	1	2
$M_0$	0	$\llbracket 0, 4 \rrbracket$	$\llbracket 0, 4 \rrbracket$		$\llbracket 1, 4 \rrbracket$	$\llbracket 2, 4 \rrbracket$
$M_1$	3	$\llbracket 3, 4 \rrbracket$	$\llbracket 3, 4 \rrbracket$	$\llbracket 3, 4 \rrbracket$	$\llbracket 3, 5 \rrbracket$	$\llbracket 3, 6 \rrbracket$
$M_2$	4	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 5 \rrbracket$	$\llbracket 4, 6 \rrbracket$
$M_3$	4	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 4 \rrbracket$	$\llbracket 4, 5 \rrbracket$	$\llbracket 4, 6 \rrbracket$

FIGURE 6.6 – Dates disponibles pour les rencontres si  $W_{\max} \leq LB_0 = 1$ .

Pour déterminer une borne inférieure, on considère séparément chaque participant et on cherche à savoir s'il est possible d'affecter un créneau à chacune de ses rencontres. Dans le cas présent, si on considère l'homme  $M_1$ , le graphe associé à ses rencontres est celui de la figure 6.7. Un couplage maximum entre ses rencontres et ses créneaux a une valeur de 4 (en gras sur la figure). Il manque un créneau pour que l'homme  $M_1$  puisse effectuer ses rencontres. En



FIGURE 6.7 – Rencontres de l'homme  $M_1$  et couplage maximum associé.

créant le graphe de l'homme  $M_2$ , on se rend compte qu'il manque deux créneaux pour que celui-ci puisse effectuer ses rencontres. Une borne inférieure est donc  $LB_2 = LB_0 + 2 = 3$  dans le cas présent.

Dans la figure 6.8, on présente une solution optimale pour cette instance.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1	1	2	$W_i^{(1)}$
$M_0$	0	0	1		2	3	0
$M_1$	3	3	4	5	6	7	0
$M_2$	4	4	5	6	7	8	0
$M_3$	4	5	6	4	8	9	1
	$W_j^{(2)}$	2	3	3	4	4	$W_{\max}^* = 4$

FIGURE 6.8 – Solution optimale.

**Proposition 6.4.2.** *La borne inférieure  $LB_2$  domine toujours la borne inférieure  $LB_1$ .*

*Démonstration.* 6.4.2 Dans le calcul de la borne inférieure  $LB_1$ , on relâche les contraintes de disjonction des participants puis on résout séparément le problème de minimisation de chacun des participants. Pour un homme  $M_i$ , minimiser ses attentes de cette manière est équivalent au problème suivant : on définit un graphe biparti dont les sommets représentent les femmes que doit rencontrer  $M_i$  et les créneaux de  $\llbracket r_i^{(1)}, e_i^{(1)} + LB_0 \rrbracket$  (on ajoute toujours  $LB_0$  pour éviter les intervalles vides). Ce graphe est noté  $\mathcal{G}_{M_i}$ . Une arête relie une femme  $F_j$  à l'ensemble des créneaux  $\llbracket r_{ij}, e_i^{(1)} - 1 + LB_0 \rrbracket$ . On calcule le cardinal d'un couplage maximum dans ce graphe. La différence entre le nombre de femmes que doit rencontrer  $M_i$  et ce cardinal plus  $LB_0$  fournit le nombre minimum d'attentes de  $M_i$  qui permet de calculer  $LB_1$ .

Le calcul de  $LB_2$  est similaire, cependant, dans le graphe  $G_{M_i}$  de la proposition-définition 6.4.1, l'ensemble des arêtes qui sont reliées à  $F_j$  est  $\llbracket r_{ij}, d_{ij} - 1 + LB_0 \rrbracket = \llbracket r_{ij}, \min(e_i^{(1)}, e_j^{(2)}) - 1 + LB_0 \rrbracket \subset \llbracket r_{ij}, e_i^{(1)} - 1 + LB_0 \rrbracket$ . Le graphe  $G_{M_i}$  est donc un sous-graphe de  $\mathcal{G}_{M_i}$ . Le cardinal d'un couplage maximum dans le graphe biparti (qu'on a noté  $G_{M_j}$  dans le cadre du calcul de  $LB_2$ ) est donc inférieur ou égal à celui calculé pour le graphe  $\mathcal{G}_{M_i}$  dans la recherche de  $LB_1$ , ce qui implique que  $LB_1 \leq LB_2$ .  $\square$

## 6.5 Condition nécessaire de non attente des participants $LB_3$

Dans cette section, on utilise une approche différente par rapport aux autres bornes définies jusqu'à présent. On relâche ici les contraintes d'unicité de date de chaque rencontre entre les dates 0 et  $e_{\min} - 1$ . La borne que nous allons définir est basée sur une approche destructive : on impose que l'objectif  $W_{\max}$  soit nul et on construit indépendamment une affectation des rencontres de chaque date  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$ . On part du constat que si aucun participant n'attend

durant la soirée ( $W_{\max} = 0$ ), alors il est nécessaire qu'à chaque créneau tous les participants présents soient occupés par une rencontre. On en déduit la proposition suivante :

**Proposition 6.5.1.** *Pour une instance donnée, on construit pour toute date  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$  le graphe biparti noté  $G_k$  dont les sommets représentent les femmes et les hommes arrivés à la date  $k$  ou avant, et une arête relie deux participants devant se rencontrer. S'il existe un ordonnancement sans attentes pour cette instance, alors pour chaque graphe  $G_k$  avec  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$ , il existe un couplage parfait.*

*En particulier, pour tout  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$ , le nombre de femmes présentes à la date  $k$  est égal au nombre d'hommes présents à la date  $k$ .*

*Démonstration.* 6.5.1 Supposons que pour une date  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$ , le graphe  $G_k$  n'admette pas de couplage parfait, alors l'un des participants ne peut pas effectuer de rencontre à cette date. Ce participant a un créneau d'attente à la date  $k$  car sa date de fin au plus tôt est supérieure ou égale à  $e_{\min}$ .  $\square$

Dans certains cas, la proposition 6.5.1 permet de déterminer si 1 est une borne inférieure de l'objectif (qu'on notera  $LB_3$ ). Il suffit de chercher pour chaque date  $k \in \llbracket 0, e_{\min} - 1 \rrbracket$  si le graphe  $G_k$  admet un couplage parfait. Si pour l'un de ces graphes, il n'existe pas de couplage parfait, alors une borne inférieure de l'objectif est 1. On peut utiliser l'algorithme de Hopcroft et Karp en  $O(n^{5/2})$  (Hopcroft et Karp, 1973) pour déterminer si un couplage parfait existe. Dans la pratique, on vérifie en premier s'il y a autant d'hommes que de femmes arrivés à chaque date avant de rechercher des couplages parfaits. Il apparaîtra dans les résultats que dans la grande majorité des cas, pour des instances à forte densité de rencontres, cette vérification est suffisante pour déterminer si  $LB_3 = 1$ .

**Exemple 6.5.1.** L'instance suivante illustre le cas où la borne  $LB_2$  est nulle (borne dominante jusqu'à présent) tandis qu'on peut facilement déterminer qu'il n'est pas possible d'ordonnancer l'ensemble des rendez-vous sans attente.

		$F_0$	$F_1$	$F_2$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1
$M_0$	0		$\llbracket 0, 1 \rrbracket$	$\llbracket 1, 1 \rrbracket$
$M_1$	0	$\llbracket 0, 0 \rrbracket$	$\llbracket 0, 2 \rrbracket$	$\llbracket 1, 2 \rrbracket$
$M_2$	2		$\llbracket 2, 2 \rrbracket$	$\llbracket 2, 3 \rrbracket$

FIGURE 6.9 – Intervalles de dates possibles si  $W_{\max} = 0$ .

À l'aide de la figure 6.9, on vérifie que  $LB_2 = 0$ . À la date zéro, chacun des participants peut effectuer une rencontre, un couplage parfait est représenté par  $(M_1, F_0)$  et  $(M_0, F_1)$ . Cependant, on ne peut pas trouver de couplage parfait entre les participants présents à la date 1. En effet, seulement deux hommes sont présents à cette date ( $M_0$  et  $M_1$ ) tandis que trois femmes sont présentes ( $F_0$ ,  $F_1$  et  $F_2$ ). On a ainsi démontré que  $LB_3 = 1$  et amélioré la meilleure borne trouvée jusqu'à présent.

Ainsi, la condition nécessaire de non attente des participants de la proposition 6.5.1 induit une borne inférieure rapide à calculer et qui complète la borne  $LB_2$ . Cependant, cette borne est au plus égale à 1 quelle que soit la taille de l'instance, elle ne peut pas être une bonne estimation de l'objectif si celui-ci est grand. Cette borne n'est donc utile qu'en complément lorsque  $LB_2$  est inefficace.

## 6.6 Résultats expérimentaux

Dans cette section, on compare les bornes inférieures décrites dans ce chapitre obtenues pour l'ensemble des 300 instances générées. On dispose de l'ensemble des solutions optimales pour chaque instance déterminées par le programme linéaire en nombres entiers présenté dans le chapitre 8.

Dans le tableau 6.1, on présente la proportion des instances pour lesquelles la borne inférieure est égale à la valeur optimale de la fonction objectif. La borne  $LB_2$  domine largement les autres. On note aussi l'intérêt de  $LB_3$  qui permet de gagner 7 optima par rapport à  $LB_2$ . Bien que la borne  $LB_3$  ne puisse pas être supérieure à la valeur 1, elle reste intéressante car elle apporte des informations complémentaires à la borne  $LB_2$  sur certaines instances dont l'objectif optimal est petit.

	$LB_0$	$LB_1$	$LB_2$	$LB_3$	$LB_{best}$
14p.	0/60	0/60	<b>47/60</b>	7/60	<b>48/60</b>
24p.	0/60	0/60	<b>44/60</b>	9/60	<b>46/60</b>
40p.	0/60	0/60	<b>42/60</b>	3/60	<b>42/60</b>
50p.	0/60	0/60	<b>45/60</b>	6/60	<b>47/60</b>
70p.	0/60	0/60	<b>39/60</b>	12/60	<b>41/60</b>
Total	0/300	0/300	<b>217/300</b>	37/300	<b>224/300</b>

Tableau 6.1 – Proportion de bornes inférieures égales à l’optimum

Dans le tableau 6.2, on présente les distances moyennes à l’optimum des différentes bornes inférieures de ce chapitre. On constate que  $LB_1$ , qui domine théoriquement  $LB_0$ , est égale à cette dernière pour la grande majorité des instances testées. La borne  $LB_2$  confirme sa dominance, elle a en effet l’avantage de rester proche de l’optimum quelle que soit la taille de l’instance.

	$LB_0$	$LB_1$	$LB_2$	$LB_3$	$LB_{best}$
14p.	2,63	2,62	<b>0,27</b>	1,80	<b>0,20</b>
24p.	3,93	3,93	<b>0,37</b>	2,95	<b>0,28</b>
40p.	6,30	6,30	<b>0,42</b>	5,30	<b>0,42</b>
50p.	7,85	7,85	<b>0,32</b>	6,85	<b>0,28</b>
70p.	9,88	9,88	<b>0,42</b>	8,88	<b>0,38</b>
Total	6,12	6,12	<b>0,36</b>	5,16	<b>0,31</b>

Tableau 6.2 – Distances moyennes à l’optimum

Le tableau 6.3 présente les proportions d’optima trouvés par proportion de participants. Les instances carrées ont un objectif généralement plus petit que les autres instances, la borne  $LB_3$  est donc plus performante sur ces instances,  $LB_3$  est égale à l’optimum dans un tiers des cas pour les instances carrées.

	$LB_0$	$LB_1$	$LB_2$	$LB_3$	$LB_{best}$
40%/60%	0/100	0/100	63/100	2/100	63/100
50%/50%	0/100	0/100	85/100	34/100	92/100
60%/40%	0/100	0/100	69/100	1/100	69/100

Tableau 6.3 – Proportion de bornes inférieures égales à l’optimum

Les temps de calcul de toutes ces bornes sont très petits. L’ordre de grandeur est le millième de seconde quelle que soit la borne calculée et ne dépasse pas 5ms.

## 6.7 Conclusion

Dans ce chapitre, nous avons présenté plusieurs méthodes de calcul de bornes inférieures du problème de speed-dating général (qui est  $\mathcal{NP}$ -difficile) basées sur la relaxation des contraintes disjonctives et des contraintes d’unicité des rendez-vous. La borne  $LB_2$  qui s’obtient par détermination de couplages maximaux a l’avantage d’être très rapide à calculer et de rester très proche de l’optimum quelle que soit la taille de l’instance. La borne  $LB_3$  ne peut fournir au mieux qu’une borne de valeur 1, elle n’est donc pas suffisante pour fournir une bonne évaluation de l’objectif. Cependant, elle peut donner d’assez bons résultats pour les instances carrées avec retard et à forte densité de rencontres sur lesquelles la borne  $LB_2$  est en difficulté, puisque ces instances ont en général un objectif très petit. La borne  $LB_3$  domine parfois la borne  $LB_2$  car cette première prend en considération l’ensemble des dates d’arrivée des participants tandis que  $LB_2$  est aveugle concernant les dates d’arrivée de l’une des populations. Coupler le calcul de  $LB_2$  à celui de  $LB_3$  est compétitif et fournit un outil puissant d’évaluation par défaut de notre problème. Les limites de la borne

inférieure  $LB_2$  viennent du fait que la planification des rendez-vous d'un participants est évaluée indépendamment des autres.

Le chapitre suivant est consacré au développement d'heuristiques de type glouton pour ce problème dans le but d'obtenir des bornes supérieures et ainsi mieux cerner la valeur optimale de l'objectif.



# Chapitre 7

## Bornes supérieures

### Sommaire

<b>7.1</b>	<b>Introduction</b>	<b>69</b>
<b>7.2</b>	<b>Bornes déterminées par des heuristiques d'ordonnancement en série</b>	<b>70</b>
<b>7.3</b>	<b>Bornes déterminées par des heuristiques d'ordonnancement en parallèle</b>	<b>71</b>
<b>7.4</b>	<b>Résultats expérimentaux</b>	<b>71</b>
<b>7.5</b>	<b>Conclusion</b>	<b>75</b>

### 7.1 Introduction

Dans le chapitre précédent, nous avons étudié des méthodes d'évaluation par défaut de notre problème de speed-dating. Nous avons ainsi déterminé des bornes inférieures pour le nombre d'attentes maximal des participants. Dans ce chapitre, nous nous intéressons à la construction de solutions réalisables qui vont nous fournir des bornes supérieures. Dans toute méthode de résolution, l'initialisation par une bonne solution est un facteur qui peut être déterminant quant à l'efficacité de cette méthode. Dans ce chapitre, nous nous consacrons à l'élaboration d'heuristiques de type glouton qui ont l'avantage d'être faciles et rapides à calculer. Celles-ci fourniront plus tard les bases d'une méthode de résolution arborescente (cf. chapitre 10).

La littérature regorge d'heuristiques pour les problèmes d'ordonnancement. On recense principalement deux types d'heuristiques de type glouton basées sur des priorités données à chacune des tâches à effectuer.

Le premier type consiste en l'affectation des tâches en série tel que proposé pour le job-shop dans (Artigues et al., 2005). L'idée est de déterminer un sous-ensemble de tâches qui peuvent être ordonnancées en premier. On classe ces tâches selon un ordre de priorité défini par un critère heuristique. La première tâche de cette liste est ordonnancée en premier. Ainsi à chaque itération, une liste de priorités est définie (celle-ci peut être statique si le critère ne dépend pas de l'ordonnancement partiel des tâches ou dynamique si cette liste doit être mise à jour à chaque itération) et une unique tâche est ordonnancée. De nombreux critères ont été définis pour les différents problèmes d'ordonnancement. Dans (Haupt, 1989), Haupt recense de nombreuses règles de priorité pour les problèmes de job shop, nombre d'entre elles sont suffisamment génériques pour être reprises dans d'autres problèmes d'ordonnancement.

Le second type consiste en l'affectation des tâches de manière parallèle. Il s'agit de déterminer la date la plus petite à laquelle peut être ordonnancée une tâche et de déterminer un ensemble éligible de tâches non ordonnancées à cette date. Ensuite, un choix est effectué sur ces tâches pour déterminer lesquelles sont ordonnancées à cette date selon un critère heuristique de priorité. Contrairement aux heuristiques d'ordonnancement en série dans lesquelles une itération correspond à l'ordonnancement d'une tâche, les heuristiques d'ordonnancement en parallèle itèrent sur le temps et à chaque itération plusieurs tâches peuvent être ordonnancées. Ce type d'heuristique a été développé, par exemple, dans (Bräsel et al., 1993) pour le problème d'open shop  $O||C_{\max}$ .

Dans ce chapitre, nous envisageons ces deux types d'approche pour résoudre notre problème. Dans la section 7.2, nous étudions des heuristiques d'ordonnancement en série basées sur des priorités données à chaque rencontre. Ensuite, dans la section 7.3, nous proposons des heuristiques d'ordonnancement en parallèle basées sur la résolution

de problèmes d'affectation successifs. Dans la section 7.4, nous présentons l'ensemble des résultats obtenus par ces heuristiques et les comparons avec les résultats de bornes inférieures que nous avons obtenus dans le chapitre 6.

## 7.2 Bornes déterminées par des heuristiques d'ordonnement en série

Les différents algorithmes présentés dans cette section sont basés sur une liste de priorité des rencontres qui vont être placées en série. Pour chaque date, on ordonne les rencontres disponibles à cette date selon des critères prédéfinis puis on affecte les rencontres prioritaires dans cette liste. À la fin de cette étape, on met éventuellement à jour la liste des priorités sur les rencontres restantes. Toutes les heuristiques de ce type fournissent un ordonnancement actif, c'est-à-dire qu'aucune rencontre ne peut être placée plus tôt sans retarder d'autres rencontres. Les ordonnancements actifs sont dominants pour les critères réguliers, ce qui est notre cas car l'objectif  $W_{\max}$  est croissant en les dates de fin des participants. L'algorithme 4 présente la trame générale des heuristiques utilisées.

---

### Algorithme 4 Heuristique d'ordonnement en série

---

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}, (r_j^{(2)})_{j \in \llbracket 0, n-1 \rrbracket}, (U \cup V, E)$ .

```

1:  $t := 0$ 
2: Tant que il existe des rencontres non planifiées faire
3:    $L :=$  liste des rencontres non ordonnancées et disponibles à la date  $t$ .
4:   Trier  $L$  par ordre de priorité.
5:   Tant que  $L \neq \emptyset$  faire
6:     Planifier la première rencontre  $meet$  de  $L$  à la date  $t$ .
7:     Retirer  $meet$  de  $L$  ainsi que les rencontres faisant intervenir un participant de  $meet$ .
8:   Fin Tant que
9:    $t := t + 1$ 
10: Fin Tant que
11: Calculer  $W_{\max}$ .
```

---

Le tri de la ligne 4 se fait selon l'une des différentes règles de priorité détaillées ici.

- $r$  : ordre lexicographique des dates d'arrivée des femmes puis des hommes.
- $m$  : sont prioritaires les rencontres dont les participants ont le plus de rencontres non planifiées. En effet, les participants ayant le plus de rencontres à planifier sont susceptibles d'avoir des créneaux d'attentes s'ils ne sont pas planifiés prioritairement. Ce critère est l'adaptation du critère classique LPT (Longest Processing Time).
- $d$  : les rencontres sont classées par ordre croissant de dates d'échéance des rencontres  $d_{ij}$ . Il est à noter que cet ordre est indépendant de l'affectation partielle des rencontres. Il n'est pas nécessaire de mettre à jour la liste de priorité car celle-ci est calculée à l'initialisation de l'algorithme. Ce critère est l'adaptation du critère classique EDD (Earliest Due Date).
- $w$  : sont prioritaires les rencontres dont les participants ont le plus de créneaux d'attente déjà planifiés. Ce critère est directement corrélé à l'objectif de minimisation des attentes des participants.

On obtient ainsi les heuristiques notées  $H_r, H_m, H_w$  et  $H_d$ . Les critères utilisés ne fournissent pas un ordre total sur les rencontres. À chaque itération, de nombreuses rencontres ont la même priorité. Dans ce cas, on choisit la première rencontre parmi celles dont la date de disponibilité est la plus petite. De plus, de manière à choisir plus judicieusement la rencontre à affecter à chaque itération, on combine ces critères de manière lexicographique dans le but d'obtenir des relations d'ordre plus fines. On note par exemple  $H_{wmd}$  l'heuristique obtenue en considérant dans l'ordre lexicographique les critères  $w, m$  et  $d$ . Chacune des heuristiques obtenues par permutation de ces trois critères sera étudiée dans la section 7.4.1 qui présente les résultats expérimentaux. Par défaut, on ajoute systématiquement le critère  $r$  en dernier.

Les heuristiques en série ont l'avantage d'être rapides pour déterminer une solution valide. Cependant, elles ne prennent qu'une décision à chaque itération et sont donc aveugles sur l'ensemble des rencontres restantes. Afin de pallier cet inconvénient, nous développons des heuristiques d'ordonnement en parallèle présentées dans la section suivante qui prennent plusieurs décisions simultanément et ont donc une vision plus globale.

### 7.3 Bornes déterminés par des heuristiques d'ordonnancement en parallèle

Nous proposons différentes heuristiques basées sur la résolution de problèmes d'affectation. À chaque date, l'idée est de déterminer le nombre maximal de rencontres planifiables simultanément parmi celles disponibles. Chaque rencontre possède un poids correspondant à la priorité qu'on lui donne vis-à-vis d'un critère heuristique. Parmi l'ensemble des couplages de cardinal maximal, on en choisit un qui minimise (ou maximise) le coût des rencontres. Il s'agit d'un problème d'affectation. Ces heuristiques ont une vision plus globale de l'ensemble du planning, puisqu'elles ordonnent plusieurs rencontres simultanément. Elles restent néanmoins aveugles d'une date à l'autre. L'algorithme 5 présente la trame générale des heuristiques d'ordonnancement en parallèle utilisées.

---

**Algorithme 5** Heuristique de couplage
 

---

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}, (r_j^{(2)})_{j \in \llbracket 0, n-1 \rrbracket}, (U \cup V, E)$

- 1:  $t := 0$
- 2: **Tant que** il existe des rencontres non planifiées **faire**
- 3:    $D :=$  ensemble des rencontres non planifiées et disponibles à la date  $t$
- 4:   Mettre à jour les coûts des rencontres de  $D$
- 5:   Construire le graphe biparti valué  $G_D$  des rencontres de  $D$
- 6:   **Si**  $E \neq \emptyset$  **Alors**
- 7:     Résoudre le problème d'affectation associé au graphe  $G_D$ .
- 8:     Planifier les rencontres du couplage ainsi déterminé à la date  $t$ .
- 9:   **Fin Si**
- 10:    $t := t + 1$
- 11: **Fin Tant que**
- 12: Calculer  $W_{\max}$ .

---

À la ligne 4, le coût d'une rencontre correspond à la priorité qu'on lui donne dans l'ordonnancement. Les coûts utilisés correspondent aux critères définis dans la section précédente :

- $match_m$  ou  $match_{\Sigma m}$  : nombre maximal ou somme des nombres de rencontres restant à effectuer par les participants ;
- $match_d$  : date d'échéance de la rencontre, (la date d'échéance est une constante ne dépendant pas de l'affectation partielle des rencontres, celle-ci n'a donc pas besoin d'être mise à jour) ;
- $match_w$  ou  $match_{\Sigma w}$  : nombre maximal ou somme des attentes des deux participants.

Les heuristiques correspondantes sont notées  $H_{match_m}, H_{match_{\Sigma m}}, \text{etc.}$

Le graphe  $G_D$  de la ligne 5 est le sous-graphe des rencontres dans lequel on a uniquement conservé les arêtes des rencontres non ordonnancées des participants présents à la date  $t$  (rencontres de l'ensemble  $D$ ).

On rappelle que le problème d'affectation dans un graphe biparti est équivalent au problème de couplage maximum à coût maximal (ou minimal). Ainsi, pour chaque date, l'heuristique  $H_{match_{\Sigma m}}$  cherche le couplage maximum dont la somme des nombres de rencontres restantes à effectuer par les participants est maximal. L'algorithme hongrois développé par Kuhn (Kuhn, 1955) résout ce problème en  $O(n^4)$ , où  $n$  est le nombre de sommets du graphe. Une version améliorée de l'algorithme par Edmond et Karp fait tomber la complexité de ce problème en  $O(n^3)$  (Edmonds et Karp, 1972).

### 7.4 Résultats expérimentaux

Nous avons testé les heuristiques décrites dans ce chapitre sur l'ensemble des instances générées pour notre problème. Les résultats obtenus sont présentés dans cette section. Nous disposons de l'ensemble des solutions optimales de ces instances (elles seront déterminées dans le chapitre 8 consacré à des modèles de programmation linéaire en nombres entiers).



### 7.4.1 Bornes déterminées par des heuristiques d'ordonnancement en série

Le tableau 7.1 présente la proportion d'optima trouvés par les heuristiques de liste sur l'ensemble des 300 instances générées. Ces résultats sont présentés de manière à ordonner par ligne les relations d'ordre des moins fines aux plus fines. On constate que les heuristiques  $w$  et  $wdm$  sont celles qui permettent d'obtenir le nombre maximal de solutions optimales dans leur catégorie. Le critère  $w$  (participants qui ont le plus grand nombre d'attentes) est a priori pertinent puisqu'il correspond à l'objectif de minimisation du plus grand nombre de créneaux d'attentes  $W_{\max}$ .

On remarque que les heuristiques dont le premier critère est  $m$  (rencontres des participants qui ont le plus grand nombre de rencontres non affectées) sont très mauvaises car l'objectif  $W_{\max}$  n'est pas nécessairement corrélé aux nombre de rencontres restantes pour les participants. Cependant, ce critère permet de discriminer des rencontres qui ne l'ont pas été par les autres et est donc très intéressant à positionner en dernier. Par exemple, l'heuristique  $H_{wdm}$  trouve 11 instances optimales de plus que  $H_{wd}$ .

$r$	$m$		$d$		$w$	
39/300	13/300		49/300		<b>63/300</b>	
	$md$	$mw$	$dm$	$dw$	$wm$	$wd$
	14/300	17/300	<b>72/300</b>	32/300	48/300	64/300
	$mdw$	$mwd$	$dmw$	$dwm$	$wmd$	$wdm$
	17/300	17/300	70/300	50/300	48/300	<b>75/300</b>

Tableau 7.1 – Proportion d'optima trouvés par les heuristiques de liste

Le nombre important de solutions optimales trouvées par l'heuristique  $dm$  trouve son explication dans le tableau 7.2 qui présente le nombre de solutions optimales trouvées par proportion de participants (toutes tailles d'instances confondues). En effet, les heuristiques de type  $dm$  et  $dmw$  dominent pour les instances dont le nombre de femmes est égal au nombre d'hommes. Pour ce type d'instances, la fonction objectif est assez petite (en moyenne 2), contrairement aux instances non carrées qui ont en moyenne un objectif égal à 20% de la taille de l'instance (nombre de femmes + nombre d'hommes). Ainsi, le critère  $w$  ne permet pas de discriminer suffisamment les rencontres à placer en priorité au début de l'événement dans les instances carrées car les participants ont des nombres d'attente petits et proches, alors que le critère  $d$  (placer en priorité les rencontres qui ont la plus petite date d'échéance) ne dépend pas de l'instantiation partielle et est donc plus efficace si les participants ont des nombres d'attente proches. À noter que l'heuristique  $wdm$  reste compétitive vis-à-vis des autres heuristiques sur les instances carrées.

Prop F/H	$r$	$m$		$d$		$w$	
40%/60%	18/100	3/100		21/100		<b>31/100</b>	
50%/50%	9/100	7/100		14/100		<b>15/100</b>	
60%/40%	12/100	3/100		14/100		<b>17/100</b>	
		$md$	$mw$	$dm$	$dw$	$wm$	$wd$
40%/60%		5/100	6/100	19/100	13/100	22/100	<b>26/100</b>
50%/50%		6/100	9/100	<b>35/100</b>	11/100	16/100	16/100
60%/40%		3/100	2/100	18/100	8/100	12/100	<b>22/100</b>
		$mdw$	$mwd$	$dmw$	$dwm$	$wmd$	$wdm$
40%/60%		6/100	6/100	18/100	16/100	22/100	<b>31/100</b>
50%/50%		9/100	9/100	<b>34/100</b>	16/100	14/100	22/100
60%/40%		2/100	2/100	18/100	8/100	12/100	<b>22/100</b>

Tableau 7.2 – Proportion d'optima trouvés par les heuristiques de liste par proportion d'hommes et de femmes

Afin de préciser ces résultats, les moyennes des distances absolues et relatives à l'optimum des objectifs déterminés par les heuristiques sont présentées dans le tableau 7.3 en fonction de la proportion des participants. Pour rappel, la distance relative est la valeur suivante  $(UB - Opt)/Opt$  où  $UB$  est la valeur de l'objectif déterminée par l'heuristique et  $Opt$  est la valeur optimale de l'objectif (la valeur optimale de l'objectif n'est jamais nulle pour l'ensemble de nos instances). Dans un souci de lisibilité, on ne présente que les trois meilleures heuristiques. On remarque que

les résultats des trois heuristiques sont similaires sur les instances carrées. Si les distances relatives sont si grandes pour les instances carrées, c'est parce que les valeurs optimales de l'objectif sont très petites (entre 1 et 5 pour une moyenne de 2,12).

Prop F/H	$dm$		$dmw$		$wdm$	
	dist. abs.	dist. rel.	dist. abs.	dist. rel.	dist. abs.	dist. rel.
40%/60%	5,1	59,9%	5,1	59,8%	<b>1,0</b>	<b>15,7%</b>
50%/50%	<b>1,2</b>	94,5%	1,3	98,8%	1,3	<b>91,7%</b>
60%/40%	5,0	52,8%	5,0	53,5%	<b>1,2</b>	<b>17,3%</b>

Tableau 7.3 – Moyennes des distances absolues et relatives à l'optimum

Le tableau 7.4 présente les proportions d'optima trouvées par les heuristiques d'ordonnement en série en fonction de la taille des instances. De manière générale, on peut noter que toutes heuristiques confondues, la difficulté à trouver une solution optimale augmente avec la taille de l'instance. Pour  $H_{wdm}$ , on passe de plus de 60% des instances résolues à l'optimum pour les instances à 14 participants à moins de 10% pour les instances à 70 participants.

Taille	$r$	$m$	$d$		$w$		
14 p.	34/60	11/60	<b>43/60</b>		31/60		
24 p.	5/60	1/60	6/60		<b>15/60</b>		
40 p.	0/60	0/60	0/60		<b>8/60</b>		
50 p.	0/60	1/60	0/60		<b>9/60</b>		
70 p.	0/60	0/60	0/60		0/60		
		$md$	$mw$	$dm$	$dw$	$wm$	$wd$
14 p.		12/60	13/60	<b>42/60</b>	24/60	27/60	32/60
24 p.		2/60	2/60	<b>20/60</b>	8/60	8/60	17/60
40 p.		0/60	1/60	5/60	0/60	6/60	<b>7/60</b>
50 p.		0/60	1/60	4/60	0/60	6/60	<b>7/60</b>
70 p.		0/60	0/60	<b>1/60</b>	0/60	<b>1/60</b>	<b>1/60</b>
		$mdw$	$mwd$	$dmw$	$dwm$	$wmd$	$wdm$
14 p.		13/60	13/60	<b>41/60</b>	36/60	27/60	36/60
24 p.		2/60	2/60	21/60	14/60	8/60	<b>25/60</b>
40 p.		1/60	1/60	4/60	0/60	<b>6/60</b>	<b>6/60</b>
50 p.		1/60	1/60	4/60	0/60	<b>6/60</b>	3/60
70 p.		0/60	0/60	0/60	0/60	1/60	<b>5/60</b>

Tableau 7.4 – Proportion d'optima trouvés par les heuristiques de liste par taille d'instance

Le tableau 7.5 présente les moyennes des distances absolues et relatives à l'optimum des objectifs déterminés par les meilleures heuristiques de liste en fonction de la taille des instances. La qualité de la solution trouvée par une heuristique est également corrélée à sa taille. Cela confirme que  $H_{wdm}$  est plus efficace que les autres heuristiques.

Taille	$dm$		$dmw$		$w$		$wdm$	
	dist. abs.	dist. rel.	dist. abs.	dist. rel.	dist. abs.	dist. rel.	dist. abs.	dist. rel.
14p.	<b>0,4</b>	23,3%	<b>0,4</b>	25,8%	0,6	33,5%	<b>0,4</b>	<b>22,6%</b>
24p.	1,4	56,1%	1,4	54,7%	1,0	51,1%	<b>0,7</b>	<b>32,1%</b>
40p.	3,6	63,0%	3,7	65,6%	1,7	55,1%	<b>1,3</b>	<b>36,9%</b>
50p.	5,1	79,2%	5,0	80,8%	1,9	77,4%	<b>1,6</b>	<b>46,1%</b>
70p.	8,5	123,6%	8,5	126,5%	2,6	116,1%	<b>1,9</b>	<b>70,2%</b>

Tableau 7.5 – Différences moyennes absolues et relatives pour les heuristiques de liste

Les temps de calcul de l'ensemble de ces heuristiques sont très courts. L'ordre de grandeur est le centième de seconde.

### 7.4.2 Bornes déterminées par heuristiques d'ordonnancement en parallèle

Dans cette section, nous présentons les résultats obtenus pour les heuristiques de couplage. Nous rappelons que pour les critères  $w$  et  $m$ , les poids des rencontres sont définis de deux manières différentes. Dans le cas noté  $match_{\Sigma m}$  (resp.  $match_{\Sigma w}$ ), on considère la somme des nombres de rencontres non affectées aux participants (resp. des attentes des participants). Dans le cas noté  $match_m$  (resp.  $match_w$ ), on considère le maximum des rencontres non affectées aux participants (resp. des attentes des participants). Pour l'heuristique  $match_d$ , le poids considéré est la date d'échéance de la rencontre.

Le tableau 7.6 présente les nombres d'optima déterminés par les heuristiques de couplage par proportion de femmes et d'hommes. De la même manière que pour les heuristiques de liste, le critère  $w$  est meilleur pour les instances non carrées tandis que le critère  $d$  permet d'obtenir de meilleures solutions pour les instances carrées. Enfin, on remarque la plus grande efficacité des heuristiques de couplages par rapport aux heuristiques de liste : l'heuristique  $match_{\Sigma w}$  trouve 35,67% des optima quand l'heuristique de liste  $wdm$  en détermine 25%.

Prop F/H	$match_m$	$match_{\Sigma m}$	$match_d$	$match_w$	$match_{\Sigma w}$
40%/60%	15/100	6/100	13/100	<b>36/100</b>	37/100
50%/50%	32/100	10/100	<b>45/100</b>	29/100	34/100
60%/40%	14/100	8/100	6/100	35/100	<b>36/100</b>
Total	61/300	24/300	64/300	100/300	<b>107/300</b>

Tableau 7.6 – Proportion d'optima trouvés par les heuristiques de couplage par proportion femmes/hommes

Le tableau 7.7 présente les moyennes des distances absolues et relatives à l'optimum trouvées par les heuristiques de couplage. On constate la nette amélioration des solutions trouvées par les heuristiques de couplage par rapport à celles de liste. Le critère  $d$  reste le plus efficace pour les instances carrées de même que le critère  $w$  pour les instances non carrées.

Prop F/H	$match_m$		$match_{\Sigma m}$		$match_d$		$match_w$		$match_{\Sigma w}$	
	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.
40%/60%	2,0	30,4%	2,5	42,9%	5,4	70,2%	<b>1,0</b>	<b>15,4%</b>	<b>1,0</b>	16,0%
50%/50%	1,0	68,9%	2,7	183,9%	<b>0,7</b>	<b>49,2%</b>	1,4	99,9%	1,2	87,3%
60%/40%	2,1	26,7%	2,3	34,7%	5,6	62,7%	<b>0,8</b>	11,7%	<b>0,8</b>	<b>11,5%</b>

Tableau 7.7 – Distances moyennes à l'optimum pour les heuristiques de couplage

Le tableau 7.8 présente les proportions d'optima déterminées par les heuristiques de couplage en fonction de la taille des instances et le tableau 7.9 présente les moyennes des distances absolues et relatives à l'optimum des objectifs définis par les heuristiques de couplage en fonction de la taille des instances. Les heuristiques à critère  $w$  dominent l'ensemble des autres heuristiques de liste et de couplage.

Taille	$match_m$	$match_{\Sigma m}$	$match_d$	$match_w$	$match_{\Sigma w}$
14 p.	35/60	19/60	30/60	<b>48/60</b>	<b>48/60</b>
24 p.	13/60	3/60	15/60	22/60	<b>27/60</b>
40 p.	6/60	1/60	11/60	15/60	<b>18/60</b>
50 p.	5/60	1/60	6/60	<b>11/60</b>	<b>11/60</b>
70 p.	2/60	0/60	2/60	<b>4/60</b>	3/60

Tableau 7.8 – Proportion d'optima trouvés par les heuristiques de couplage par taille

Taille	$match_m$		$match_{\Sigma m}$		$match_d$		$match_w$		$match_{\Sigma w}$	
	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.	dist.abs.	dist.rel.
14 p.	0,5	26,4%	0,9	47,4%	0,8	40,6%	<b>0,2</b>	13,8%	<b>0,2</b>	<b>12,9%</b>
24 p.	1,3	42,1%	2,0	75,7%	1,7	51,9%	0,8	37,6%	<b>0,7</b>	<b>32,3%</b>
40 p.	2,0	42,7%	2,9	85,2%	3,8	55,0%	<b>1,0</b>	33,2%	<b>1,0</b>	<b>31,4%</b>
50 p.	2,1	<b>43,2%</b>	3,0	95,2%	5,0	65,9%	<b>1,4</b>	53,8%	<b>1,4</b>	46,1%
70 p.	2,6	<b>55,5%</b>	3,8	132,3%	8,1	90,1%	1,8	73,5%	<b>1,7</b>	68,6%

Tableau 7.9 – Différences absolues et relatives moyennes pour les heuristiques de couplage

Parmi toutes les heuristiques présentées dans ce chapitre aucune ne domine systématiquement les autres. Si on lance l'ensemble des heuristiques et qu'on garde systématiquement la meilleure solution, on obtient 172 solutions optimales sur 300, la distance moyenne entre la meilleure borne supérieure trouvée et l'optimum est de 0,49 la moyenne de la distance relative est de 17%.

Les temps de calcul de l'ensemble de ces heuristiques sont également très courts et ont le même ordre de grandeur que les heuristiques en série (centième de seconde).

### 7.4.3 Comparaison bornes inférieures/bornes supérieures

Chacune des heuristiques de ce chapitre fournit une borne supérieure de l'objectif. On note  $UB_{best}$  la meilleure borne parmi toutes celles trouvées par les heuristiques développées dans ce chapitre. Le tableau 7.10 présente les différences entre les meilleures bornes supérieures et inférieures obtenues pour chaque instance. On remarque ainsi qu'on résout 134/300 instances, soit 44%, à l'optimum en calculant les meilleures bornes inférieures et supérieures. Les différences entre les bornes supérieures et inférieures sont petites de manière générale. On a ainsi une assez bonne évaluation de l'objectif par bornes inférieures et supérieures combinées.

$UB_{best} - LB_{best}$	0	1	2	3	4
14 p.	<b>46/60</b>	14/60			
24 p.	<b>34/60</b>	20/60	6/60		
40 p.	<b>26/60</b>	16/60	16/60	2/60	
50 p.	18/60	<b>29/60</b>	11/60	2/60	
70 p.	10/60	22/60	<b>23/60</b>	4/60	1/60
Total	<b>134/300</b>	101/300	56/300	8/300	1/300

Tableau 7.10 – Différences entre les meilleures bornes supérieures et inférieures.

## 7.5 Conclusion

Dans ce chapitre, nous avons développé deux types d'heuristiques pour obtenir des bornes supérieures. Ces heuristique sont basées sur la mise en place de priorités sur les rencontres. Le premier type prend en compte une liste de priorité dynamique qui indique l'ordre dans lequel affecter les rencontres en série. Le second affecte en parallèle plusieurs rencontres à une date fixée en favorisant les rencontres prioritaires par la résolution de problèmes d'affectation.

Les résultats obtenus montrent que les heuristiques de couplage procurent les meilleurs résultats. Ceci peut s'expliquer par le fait que lors de l'affectation des rencontres à une date, cette méthode a une vision plus globale que dans le cas de la liste de priorité puisqu'elle affecte plusieurs rencontres en même temps. C'est efficace pour notre problème car les durées des rencontres sont unitaires, il n'y a pas de précédences entre les rencontres et beaucoup de rencontres sont éligibles à une date donnée. Elles restent cependant aveugles d'une date à l'autre. Ensuite, le critère le plus efficace de priorités des rencontres est celui qui donne priorité aux participants attendant le plus. Cela n'est pas surprenant non plus puisqu'il s'agit du critère le mieux corrélé à l'objectif étudié qui est le nombre maximal d'attentes. Des résultats intéressants sont apparus pour les instances carrées : en effet, le critère de plus petite date d'échéance présente de bons résultats pour les instances dont le nombre de femmes est égal au nombre d'hommes.

Il est à noter que les instances carrées ont une valeur optimale de leur objectif très petite (2,12 en moyenne pour une valeur maximale de 5), on peut donc supposer que dans ce cas, lors de la construction d'une solution réalisable, les valeurs des attentes des participants sont très petites et souvent proches les unes des autres. Le critère de participants ayant le plus d'attentes à ce moment de la soirée ne discrimine donc pas suffisamment les rencontres à qui donner la priorité, tandis que la date d'échéance prévoit mieux les rencontres qui risquent d'engendrer des attentes.

Dans la partie suivante, nous développons plusieurs méthodes de résolution du problème de speed-dating. Les bornes inférieures et les heuristiques que nous avons décrites ici seront largement exploitées dans ces méthodes. En particulier les heuristiques fourniront des bornes supérieures utiles à l'initialisation des méthodes.

*Les bornes inférieures et supérieures de cette partie ont été présentées lors du Workshop MAPSP 2013 (Le Roux et al., 2013a).*

**Quatrième partie**

**Modélisation et résolution**



## Chapitre 8

# Programmation linéaire en nombres entiers

### Sommaire

<a href="#">8.1</a>	<a href="#">Introduction</a>	<a href="#">79</a>
<a href="#">8.2</a>	<a href="#">Modèle indexé sur le temps</a>	<a href="#">79</a>
<a href="#">8.3</a>	<a href="#">Extension du modèle indexé sur le temps</a>	<a href="#">81</a>
<a href="#">8.4</a>	<a href="#">Modèle basé sur les dates de rencontres</a>	<a href="#">83</a>
<a href="#">8.5</a>	<a href="#">Règles de dominance</a>	<a href="#">84</a>
<a href="#">8.6</a>	<a href="#">Résultats expérimentaux</a>	<a href="#">93</a>
<a href="#">8.7</a>	<a href="#">Considérations sur la génération de colonnes</a>	<a href="#">97</a>
<a href="#">8.8</a>	<a href="#">Conclusion</a>	<a href="#">98</a>

## 8.1 Introduction

Une première approche possible pour résoudre un problème d’ordonnancement  $\mathcal{NP}$ -difficile consiste à produire une modélisation du problème par un programme linéaire en nombres entiers (PLNE). Cette formalisation des contraintes sous forme d’un PLNE permet de mettre en place une première résolution à l’aide d’un solveur existant pour des instances de taille limitée. La difficulté vient de la formulation du modèle. De manière générale, on cherche un modèle dont la relaxation continue est la plus proche possible de l’enveloppe convexe des solutions entières du problème afin de converger plus vite vers une solution entière optimale. Dans ce chapitre, nous proposons différents programmes linéaires en nombres entiers pour résoudre le problème de speed-dating. Pour ces modèles, nous proposons des inégalités valides et des règles de dominance dans le but de diminuer l’espace des solutions. Nous comparons de manière empirique les différents modèles proposés dans ce chapitre.

Dans la section 8.2, nous proposons un modèle avec variables indexées sur le temps et des inégalités valides pour ce modèle. Ce modèle est étendu dans la section 8.3 à l’aide de variables additionnelles qui permettent de poser de nouvelles inégalités valides. Dans la section 8.4, nous proposons une approche différente avec un modèle disjonctif et des inégalités valides pour celui-ci. La section 8.5 est consacrée à l’étude de propriétés de notre problème qui fournissent des règles de dominance applicables aux modèles de PLNE. Enfin, nous présentons les résultats expérimentaux obtenus dans la section 8.6.

## 8.2 Modèle indexé sur le temps

Une première manière de modéliser notre problème est d’introduire des variables indexées sur le temps. Cette approche est courante pour de nombreux problèmes tels que le problème d’emploi du temps général. À ce titre, on peut citer (Even et al., 1976) et (de Werra, 1988) qui proposent une modélisation du problème d’emploi du temps général par un programme linéaire en nombres entiers indexé sur le temps. Dans ce modèle, la variable binaire  $f_{i,j,h}$  est égale



à 1 si et seulement si le professeur  $i$  enseigne à la classe  $j$  durant la période  $h$ . Nous nous inspirons de ce type de modélisation dans cette section.

### 8.2.1 Modèle de base

Le premier modèle proposé repose sur des variables binaires indicées sur les deux populations de participants et les créneaux de la soirée. Dans ce modèle, nous nous servons d'une borne inférieure et d'une borne supérieure de l'objectif notées respectivement  $LB$  et  $UB$ . Il est possible de choisir celles-ci parmi les différentes bornes décrites dans les chapitres 6 et 7. Par exemple, la dernière date possible pour un rendez-vous est  $s = d_{\max} + UB$  (où  $d_{\max}$  est la plus grande date d'échéance des rencontres), donc  $\mathcal{S} = \llbracket 0, d_{\max} + UB - 1 \rrbracket$ . Les variables que nous utilisons sont les suivantes :

- pour  $i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket$  et  $k \in \mathcal{S}$ ,  $x_{ij}^k := \begin{cases} 1 & \text{si l'homme } M_i \text{ rencontre la femme } F_j \text{ à la date } k, \\ 0 & \text{sinon.} \end{cases}$
- La variable  $W_{\max}$  est le nombre maximum de créneaux d'attente des participants.

$$\min W_{\max} \quad (8.1)$$

sous les contraintes :

$$\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, \quad \sum_{k=0}^{r_{ij}-1} x_{ij}^k = 0 \quad (8.2)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall j \in \llbracket 0, n-1 \rrbracket, \quad \sum_{k=r_{ij}}^{d_{ij}+UB-1} x_{ij}^k = p_{ij} \quad (8.3)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall k \in \llbracket r_i^{(1)}, s-1 \rrbracket, \quad \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k \leq 1 \quad (8.4)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \forall k \in \llbracket r_j^{(2)}, s-1 \rrbracket, \quad \sum_{\substack{i \in \llbracket 0, m-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k \leq 1 \quad (8.5)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall j \in \llbracket 0, n-1 \rrbracket \text{ tels que } p_{ij} = 1, \quad \sum_{k=\max(r_{ij}, d_{ij}+LB)}^{\min(d_{ij}+UB-1, s-1)} (k+1)x_{ij}^k - d_{ij} \leq W_{\max} \quad (8.6)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall j \in \llbracket 0, n-1 \rrbracket, \forall k \in \mathcal{S}, \quad x_{ij}^k \in \{0, 1\}, W_{\max} \in \llbracket LB, UB \rrbracket \quad (8.7)$$

L'objectif (8.1) est de minimiser le nombre d'attentes maximum des participants. Les contraintes (8.2) assurent qu'aucune rencontre n'est effectuée avant l'arrivée des deux participants concernés. Les contraintes (8.3) imposent que les rencontres prévues ont lieu exactement une fois alors que les rencontres interdites n'ont jamais lieu ; la somme s'arrête à  $k = d_{ij} + UB - 1$  car si la rencontre entre  $M_i$  et  $F_j$  a lieu après cette date, l'objectif  $W_{\max}$  dépassera la borne supérieure  $UB$ . Les contraintes (8.4) et (8.5) assurent que chaque participant ne peut pas effectuer plus d'une rencontre par créneau. Les contraintes (8.6) permettent de déterminer la valeur de l'objectif  $W_{\max}$  en fonction des variables  $x_{ij}^k$ .

Ces contraintes proviennent de la formule établie dans la proposition 6.2.1 et du fait que  $C_{ij} = \sum_{k=r_{ij}}^{s-1} (k+1)x_{ij}^k$ . Dans ces contraintes, la somme commence à  $k = \max(r_{ij}, d_{ij} + LB)$ . En effet, si la rencontre entre  $M_i$  et  $F_j$  a lieu avant la date  $d_{ij} + LB$ , le membre de gauche de l'inégalité est inférieur à  $LB$  ce qui est redondant car  $LB \leq W_{\max}$  par définition.

*Remarque 8.2.1.* Il est possible de scinder les contraintes (8.6) de la manière suivante :

$$\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket \text{ t.q. } p_{ij} = 1, \forall k \in \llbracket \max(r_{ij}, d_{ij}+LB), \min(d_{ij}+UB-1, s-1) \rrbracket, (k+1)x_{ij}^k - d_{ij} \leq W_{\max}. \quad (8.8)$$

Nous avons écarté cette solution pour deux raisons. D'une part, il est facile de démontrer que la relaxation continue des contraintes (8.8) est moins bonne que celle des contraintes (8.6). D'autre part, le modèle basé sur ces contraintes s'est montré moins efficace lors de tests préliminaires.

### 8.2.2 Inégalités valides pour le modèle indexé sur le temps

L'écriture des contraintes d'un modèle de programmation linéaire peut avoir une grande influence sur les performances obtenues lors de la résolution avec un solveur. Il est toujours intéressant de trouver les contraintes qui définissent la plus petite relaxation continue possible. À ces fins, nous décrivons dans cette section différentes inégalités valides additionnelles pour venir renforcer le modèle.

#### Inégalités de sac à dos mixte (KS)

On constate que si la rencontre entre un homme  $M_i$  et une femme  $F_j$  a lieu après la date  $d_{ij}$ , alors cela provoque une attente pour au moins l'un des deux participants. De manière plus précise, si cette rencontre a lieu à la date  $k + d_{ij}$  où  $k \in \mathbb{N}^*$ , alors l'un des participants a au moins  $k + 1$  créneaux d'attente. On en déduit les inégalités valides suivantes :

$$\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket \text{ tels que } p_{ij} = 1, \forall k \in \llbracket LB, UB-1 \rrbracket, \quad (k+1)x_{ij}^{d_{ij}+k} \leq W_{\max}. \quad (8.9)$$

D'un point de vue théorique, les auteurs de (Marchand et Wolsey, 1999) ont démontré que les inégalités (8.9) définissent des facettes du polyèdre défini par la relaxation continue des contraintes (8.6) du modèle indexé sur le temps. Ces dernières peuvent être vues comme des contraintes de sac à dos mixte de type 0-1 pour lesquelles chaque rencontre représente un sac à dos de capacité  $d_{ij} + W_{\max}$ , les objets sont les créneaux et le poids de chaque objet est la date de fin de ce créneau. Ces contraintes peuvent remplacer les contraintes (8.6).

#### Borne inférieure basée sur le nombre de créneaux inoccupés (IV)

L'idée principale des inégalités suivantes consiste à compter le nombre de créneaux inoccupés par chacun des participants entre sa date d'arrivée et sa date de fin au plus tôt plus une borne inférieure de l'objectif. Le nombre de rencontres d'un homme  $M_i \in \mathcal{M}$  entre sa date d'arrivée  $r_i^{(1)}$  et  $e_i^{(1)} + LB$  est la somme  $\sum_{k=r_i^{(1)}}^{e_i^{(1)}+LB-1} \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k$ . On en déduit les inégalités valides (8.10) et les inégalités équivalentes pour les femmes (8.11).

$$\forall i \in \llbracket 0, m-1 \rrbracket, \quad W_{\max} \geq e_i^{(1)} + LB - \sum_{k=r_i^{(1)}}^{e_i^{(1)}+LB-1} \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k - r_i^{(1)} \quad (8.10)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \quad W_{\max} \geq e_j^{(2)} + LB - \sum_{k=r_j^{(2)}}^{e_j^{(2)}+LB-1} \sum_{\substack{i \in \llbracket 0, m-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k - r_j^{(2)} \quad (8.11)$$

## 8.3 Extension du modèle indexé sur le temps

Afin de bénéficier d'informations sur les attentes de chaque participant avec plus de précision et dans le but de définir de nouvelles contraintes pouvant améliorer le modèle, on peut étendre le modèle précédent en ajoutant des participants fictifs. On ajoute deux femmes fictives notées  $F_w$  et  $F_e$  qui représentent respectivement les attentes et la fin de soirée de chaque homme. De la même manière, on ajoute deux hommes fictifs notés  $M_w$  et  $M_e$ . Dans ce modèle, on conserve

les contraintes (8.2), (8.3), (8.6), (8.7) et les contraintes (8.4) et (8.5) se réécrivent comme suit.

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall k \in \mathcal{S}, \quad \sum_{j \in \llbracket 0, n-1 \rrbracket \cup \{w, e\}} x_{ij}^k = 1 \quad (8.12)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \forall k \in \mathcal{S}, \quad \sum_{i \in \llbracket 0, m-1 \rrbracket \cup \{w, e\}} x_{ij}^k = 1 \quad (8.13)$$

Les contraintes (8.12) et (8.13) forcent chaque participant à rencontrer lors de chaque créneau un participant réel, un participant “attente” ou un participant “fin de soirée”. On remarque que les participants fictifs peuvent rencontrer plusieurs participants réels lors d’un même créneau (puisque la contrainte 8.3 ne porte pas sur ces derniers). Enfin, on ajoute les contraintes (8.14) et (8.15) qui permettent d’assurer qu’une fois qu’un participant a fini la soirée, il ne revient pas faire des rencontres.

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall k \in \llbracket r_i^{(1)}, s-1 \rrbracket, \quad x_{ie}^k \leq x_{ie}^{k+1} \quad (8.14)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \forall k \in \llbracket r_j^{(2)}, s-1 \rrbracket, \quad x_{ej}^k \leq x_{ej}^{k+1} \quad (8.15)$$

Ces contraintes doivent être complétées par les inégalités (8.16) et (8.17), ci dessous, afin de placer correctement les rencontres avec les participants fictifs. Sans ces contraintes, des participants peuvent rencontrer un participant “attente” ( $F_w$  ou  $M_w$ ) alors qu’elles/ils ont terminé leurs rencontres.

$$\forall i \in \llbracket 0, m-1 \rrbracket, \quad \sum_{k=r_j^{(2)}}^{s-1} x_{iw}^k \leq W_{\max} \quad (8.16)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \quad \sum_{k=r_i^{(1)}}^{s-1} x_{wj}^k \leq W_{\max} \quad (8.17)$$

### Inégalités valides pour l’extension du modèle indexé sur le temps (IVe)

L’ajout de variables représentant des participants fictifs permet de distinguer les créneaux d’attente des créneaux de fin de soirée des participants. Nous exploitons ces informations dans les inégalités valides suivantes dont l’idée est de trouver une borne inférieure sur le nombre de participants d’une population ayant un créneau d’attente à une date fixée.

On note  $nbM_k = \text{card}\{M_i \in \mathcal{M} \mid r_i^{(1)} \leq k\}$  et  $nbF_k = \text{card}\{F_j \in \mathcal{F} \mid r_j^{(2)} \leq k\}$  les nombres respectifs d’hommes et de femmes arrivés avant la date  $k \in \mathcal{S}$ . On note également  $nbM'_k = \text{card}\{M_i \in \mathcal{M} \mid r_i^{(1)} \leq k \leq e_i^{(1)} - 1\}$  et  $nbF'_k = \text{card}\{F_j \in \mathcal{F} \mid r_j^{(2)} \leq k \leq e_j^{(2)} - 1\}$ . Il s’agit des nombres respectifs d’hommes et de femmes arrivés après la date  $k \in \mathcal{S}$  et dont on sait qu’ils n’ont pas fini leurs rencontres. Remarquons que pour une date fixée  $k \in \mathcal{S}$ , si  $nbF_k \leq nbM'_k$ , alors il y a nécessairement  $nbM'_k - nbF_k$  hommes qui attendent à la date  $k$  parmi les hommes qui sont arrivés et qui n’ont pas fini leurs rencontres. On en déduit les contraintes (8.18) et (8.19).

$$\forall k \in \mathcal{S}, \text{ si } nbF_k < nbM'_k, \quad \sum_{\substack{i \in \llbracket 0, m-1 \rrbracket \\ r_i^{(1)} \leq k \leq e_i^{(1)} - 1}} x_{iw}^k \geq nbM'_k - nbF_k \quad (8.18)$$

$$\forall k \in \mathcal{S}, \text{ si } nbM_k < nbF'_k, \quad \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ r_j^{(2)} \leq k \leq e_j^{(2)} - 1}} x_{wj}^k \geq nbF'_k - nbM_k. \quad (8.19)$$

Notons que pour écrire ces contraintes, il est possible de s’affranchir des variables additionnelles ( $x_{iw}^k$  et  $x_{ie}^k$ ). En effet, si  $r_i^{(1)} \leq k \leq e_i^{(1)} - 1$ , alors  $x_{iw}^k = 1 - \sum_{j \in \mathcal{M}} x_{ij}^k$  car il est certain que l’homme  $M_i$  n’a pas terminé ses rencontres. Cependant, des tests préliminaires ont montré que les contraintes (8.18) et (8.19) réécrites avec les variables du modèle de base n’améliorent pas la relaxation continue du premier modèle et les temps de calculs sont moins bons. C’est pourquoi cette option a été écartée.

## 8.4 Modèle basé sur les dates de rencontres

Une autre approche classique de modélisation de problèmes d'ordonnancement consiste à employer des variables de disjonction qui précisent les précédences entre les jobs. Dans cette approche, sont introduites des variables binaires  $\delta_{ij}^k$  qui valent 1 si et seulement si le job  $i$  est ordonnancé avant le job  $j$  sur la machine  $k$ . Ce type de modèle a notamment été proposé pour le problème  $1|prec|\sum w_i C_i$  (Pinedo, 2012) ou le problème à machines parallèles  $P||\sum w_i C_i$  (Blazewicz et al., 1991).

Nous nous inspirons de ces modélisations pour proposer un modèle de notre problème basé sur les dates des rencontres ( $C_{ij}$ ) faisant appel à des variables de disjonction. Dans un premier temps, nous présentons le modèle de base puis nous proposons des inégalités valides pour ce modèle.

### 8.4.1 Modèle de base

Les variables de décision du modèle sont les suivantes :

- La variable  $W_{\max}$  est le nombre maximum de créneaux d'attente des participants.
- Pour  $i \in \llbracket 0, m-1 \rrbracket$ ,  $j \in \llbracket 0, n-1 \rrbracket$ , tels que  $p_{ij} = 1$ ,  $C_{ij}$  est la date de fin de la rencontre entre l'homme  $M_i$  et la femme  $F_j$ .

Ce modèle nécessite également les variables de disjonction suivantes :

- Pour  $j \in \llbracket 0, n-1 \rrbracket$ ,  $(i_1, i_2) \in \llbracket 0, m-1 \rrbracket^2$  tels que  $p_{i_1 j} = p_{i_2 j} = 1$  et  $i_1 < i_2$ ,  
 $\delta_{i_1 i_2}^j := \begin{cases} 1 & \text{si la femme } F_j \text{ rencontre l'homme } M_{i_1} \text{ avant l'homme } M_{i_2}, \\ 0 & \text{sinon.} \end{cases}$
- Pour  $i \in \llbracket 0, m-1 \rrbracket$ ,  $(j_1, j_2) \in \llbracket 0, n-1 \rrbracket^2$ , tels que  $p_{ij_1} = p_{ij_2} = 1$  et  $j_1 < j_2$ ,  
 $\eta_{j_1 j_2}^i := \begin{cases} 1 & \text{si l'homme } M_i \text{ rencontre la femme } F_{j_1} \text{ avant la femme } F_{j_2}, \\ 0 & \text{sinon.} \end{cases}$

On introduit également deux constantes  $m$  et  $M$  qui sont respectivement une borne inférieure de  $C_{ij} - C_{i'j'}$  et une borne supérieure de  $C_{ij} - C_{i'j'}$  pour  $M_i, M'_i \in \mathcal{M}$  et  $F_j, F'_j \in \mathcal{F}$ . On peut prendre  $\mu = -s - 1$  et  $M = s + 1$ . On peut ainsi modéliser le problème de speed-dating de la manière suivante :

$$\min W_{\max} \quad (8.20)$$

sous les contraintes :  $\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket$ , tels que  $p_{ij} = 1$ ,

$$r_{ij} + 1 \leq C_{ij} \quad (8.21)$$

$$\forall (i_1, i_2) \in \llbracket 0, m-1 \rrbracket^2, \text{ t.q. } i_1 < i_2 \text{ et } p_{i_1 j} = p_{i_2 j} = 1, \quad C_{i_1 j} - C_{i_2 j} \geq 1 + (\mu - 1)\delta_{i_1 i_2}^j \quad (8.22)$$

$$\forall (i_1, i_2) \in \llbracket 0, m-1 \rrbracket^2, \text{ t.q. } i_1 < i_2 \text{ et } p_{i_1 j} = p_{i_2 j} = 1, \quad C_{i_1 j} - C_{i_2 j} \leq -1 + (M + 1)(1 - \delta_{i_1 i_2}^j) \quad (8.23)$$

$$\forall (j_1, j_2) \in \llbracket 0, n-1 \rrbracket^2, \text{ t.q. } j_1 < j_2 \text{ et } p_{ij_1} = p_{ij_2} = 1, \quad C_{ij_1} - C_{ij_2} \geq 1 + (\mu - 1)\eta_{j_1 j_2}^i \quad (8.24)$$

$$\forall (j_1, j_2) \in \llbracket 0, n-1 \rrbracket^2, \text{ t.q. } j_1 < j_2 \text{ et } p_{ij_1} = p_{ij_2} = 1, \quad C_{ij_1} - C_{ij_2} \leq -1 + (M + 1)(1 - \eta_{j_1 j_2}^i) \quad (8.25)$$

$$C_{ij} - e_{ij} \leq W_{\max} \quad (8.26)$$

$$C_{ij} \in \llbracket 1, s \rrbracket, W_{\max} \in \llbracket LB, UB \rrbracket \quad (8.27)$$

$$\forall (i_1, i_2) \in \llbracket 0, m-1 \rrbracket^2, \text{ t.q. } i_1 < i_2 \text{ et } p_{i_1 j} = p_{i_2 j} = 1, \quad \delta_{i_1 i_2}^j \in \{0, 1\} \quad (8.28)$$

$$\forall (j_1, j_2) \in \llbracket 0, n-1 \rrbracket^2, \text{ t.q. } j_1 < j_2 \text{ et } p_{ij_1} = p_{ij_2} = 1, \quad \eta_{j_1 j_2}^i \in \{0, 1\} \quad (8.29)$$

Les contraintes (8.21) assurent qu'aucune rencontre n'est effectuée avant l'arrivée des deux participants concernés. Les contraintes (8.22) et (8.23) (resp. (8.24) et (8.25)) permettent de linéariser les contraintes de disjonction (8.30) (resp. (8.31)) qui assurent qu'aucune femme (resp. homme) n'effectue pas plus d'une rencontre par créneau.

$$\forall j \in \llbracket 0, n-1 \rrbracket, \forall (i_1, i_2) \in \llbracket 0, m-1 \rrbracket^2, \text{ t.q. } i_1 < i_2 \text{ et } p_{i_1 j} = p_{i_2 j} = 1, \quad C_{i_1 j} < C_{i_2 j} \text{ ou } C_{i_1 j} > C_{i_2 j} \quad (8.30)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, \forall (j_1, j_2) \in \llbracket 0, n-1 \rrbracket^2, \text{ t.q. } j_1 < j_2 \text{ et } p_{ij_1} = p_{ij_2} = 1, \quad C_{ij_1} < C_{ij_2} \text{ ou } C_{ij_1} > C_{ij_2} \quad (8.31)$$

Les contraintes (8.26) effectuent le lien entre l'objectif et les variables de date de fin de rencontres. La proposition 6.2.1, qui stipule que  $W_{\max} = \max\{C_{ij} - d_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$ , justifie ces inégalités.

### 8.4.2 Inégalités valides

Nous proposons ici des inégalités valides pour le modèle basé sur les dates des rencontres.

#### Inégalités ternaires (IVt)

On peut ajouter les inégalités valides suivantes pour le modèle basé sur les dates des rencontres.

$$\forall j \in \llbracket 0, n-1 \rrbracket, i_1, i_2, i_3 \in \llbracket 0, m-1 \rrbracket, \text{t.q. } i_1 < i_2 < i_3, p_{i_1j} = p_{i_2j} = p_{i_3j} = 1; 1 \leq \delta_{i_1i_2}^j + \delta_{i_2i_3}^j + (1 - \delta_{i_3i_1}^j) \leq 2 \quad (8.32)$$

$$\forall i \in \llbracket 0, m-1 \rrbracket, j_1, j_2, j_3 \in \llbracket 0, n-1 \rrbracket, \text{t.q. } i_1 < i_2 < i_3, p_{ij_1} = p_{ij_2} = p_{ij_3} = 1; 1 \leq \eta_{j_1j_2}^i + \eta_{j_2j_3}^i + (1 - \eta_{j_3j_1}^i) \leq 2. \quad (8.33)$$

En effet, si  $\delta_{i_1i_2}^j = \delta_{i_2i_3}^j = (1 - \delta_{i_3i_1}^j) = 1$ , alors la femme  $F_j$  rencontre  $M_{i_1}$  avant  $M_{i_2}$ ,  $M_{i_2}$  avant  $M_{i_3}$  et  $M_{i_3}$  avant  $M_{i_1}$ , ce qui est impossible. Si ces variables sont toutes nulles, l'ordre des rencontres est inversé et la solution reste impossible.

On peut généraliser ces contraintes pour  $k$  personnes ( $k \geq 3$ ). Cependant, le nombre de contraintes ajoutées devient prohibitif.

## 8.5 Règles de dominance

Nous introduisons ici des règles de dominance visant à limiter l'espace des solutions explorées. Une règle de dominance identifie un sous-ensemble du produit cartésien des domaines de définition des variables contenant au moins une solution optimale (Jouglet et Carlier, 2011). Dans notre cas, on considère  $\Omega = \{(x_{ij}^k, C_{ij}, W_{\max}) \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, k \in \mathcal{S}, x_{ij}^k \in \{0, 1\}, C_{ij} \in \llbracket 1, s \rrbracket, W_{\max} \in \mathbb{N}\}$ . Une règle de dominance détermine un sous-ensemble de  $\Omega$  noté  $\delta(\Omega)$  contenant au moins une solution optimale.

### 8.5.1 Rencontres à la date zéro

Les contraintes suivantes forcent certaines rencontres à se dérouler à la date zéro. Ces rencontres ne concernent donc que les participants arrivant à l'heure.

**Règle de dominance 8.5.1.** Soit  $(M_i, F_j)$  un couple de participants arrivant à l'heure et devant se rencontrer. On peut affecter au moins une rencontre parmi celles de  $M_i$  et  $F_j$  à la date zéro. On note  $\delta_0(\Omega)$  les solutions vérifiant ces contraintes.

**Exemple 8.5.1.** La figure 8.1 représente une solution optimale d'une instance avec retards et deux rencontres interdites. Dans cette instance, trois hommes ( $M_0, M_1, M_2$ ) et trois femmes ( $F_0, F_1, F_2$ ) arrivent à l'instant zéro.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	2	$W_i^{(1)}$
$M_0$	0	4	2	1	3	1
$M_1$	0	0	3	2	4	1
$M_2$	0		1		2	1
$M_3$	2	2	4	3	5	0
$M_4$	3	3	5	4	6	0
	$W_j^{(2)}$	1	1	1	0	$W_{\max}^* = 1$

FIGURE 8.1 – Solution optimale avec  $W_{\max}^* = 1$ .

Dans la solution de la figure 8.1, la règle 8.5.1 n'est pas respectée par les couples  $(M_0, F_1)$ ,  $(M_0, F_2)$  et  $(M_2, F_1)$ . Par exemple, aucune des rencontres du couple  $(M_0, F_1)$  (en gris) n'a lieu à la date zéro. Il existe cependant au moins une solution optimale pour cette instance vérifiant la contrainte proposée. Il suffit pour cela d'affecter, par exemple, la rencontre du couple  $(M_0, F_1)$  à la date zéro comme indiqué dans la figure 8.2. Cette solution reste admissible et optimale pour l'objectif  $W_{\max}$  car aucun des participants de ce couple n'a de rencontre à la date zéro dans la première solution.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	2	$W_i^{(1)}$
$M_0$	0	4	0	1	3	1
$M_1$	0	0	3	2	4	1
$M_2$	0		1		2	1
$M_3$	2	2	4	3	5	0
$M_4$	3	3	5	4	6	0
	$W_j^{(2)}$	1	1	1	0	$W_{\max}^* = 1$

FIGURE 8.2 – Solution optimale dominante.

On aurait pu également positionner une autre rencontre de  $M_0$  et une rencontre de  $M_2$  à la date zéro comme présenté dans la figure 8.3. Dans cette solution, les contraintes de rencontres à la date zéro sont bien vérifiées pour les couples  $(M_0, F_1)$  et  $(M_0, F_2)$  car le participant  $M_0$  effectue une rencontre à la date zéro. Ces contraintes sont également vérifiées par le couple  $(M_2, F_1)$  qui se rencontre à la date zéro.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_j^{(2)} \setminus r_i^{(1)}$	0	0	0	2	$W_i^{(1)}$
$M_0$	0	4	2	0	3	1
$M_1$	0	0	3	2	4	1
$M_2$	0		0		2	1
$M_3$	2	2	4	3	5	0
$M_4$	3	3	5	4	6	0
	$W_j^{(2)}$	1	1	1	0	$W_{\max}^* = 1$

FIGURE 8.3 – Autre solution optimale dominante.

On obtient les contraintes suivantes pour le modèle indexé sur le temps :

$$\forall (u, v) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, \text{ t.q. } r_u^{(1)} = r_v^{(2)} = 0 \text{ et } p_{uv} = 1, \quad x_{uv}^0 + \sum_{\substack{ii \in \llbracket 0, m-1 \rrbracket \\ r_i^{(1)} = 0 \\ i \neq u \\ p_{iv} = 1}} x_{iv}^0 + \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ r_j^{(2)} = 0 \\ j \neq v \\ p_{uj} = 1}} x_{uj}^0 \geq 1. \quad (8.34)$$

Le deuxième terme représente la rencontre de l'homme  $M_v$  avec une des femmes arrivées à l'heure et le troisième terme représente la rencontre de la femme  $F_u$  avec un des hommes arrivés à l'heure.

On peut également écrire les contraintes équivalentes pour le modèle basé sur les dates de rencontres, cependant cela nécessite d'introduire des constantes de type “grand M” en trop grand nombre pour qu'on puisse espérer y trouver un gain en terme d'efficacité de la modélisation.

**Proposition 8.5.2.** *L'ensemble  $\delta_0(\Omega)$  contient au moins une solution optimale.*

*Démonstration.* Pour une instance du problème  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max}$ , on considère une solution optimale ne vérifiant pas les contraintes (8.34). Dans ce cas, il existe un couple  $(M_i, F_j) \in \mathcal{M} \times \mathcal{F}$  arrivant à l'heure

$(r_i^{(1)} = r_j^{(2)} = 0)$  et devant se rencontrer ( $p_{ij} = 1$ ) pour lequel aucun des deux participants n'effectue de rencontre à la date zéro, c'est-à-dire :

$$x_{ij}^0 + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ r_{i'}^{(1)} = 0 \\ i' \neq i \\ p_{i'j} = 1}} x_{i'j}^0 + \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ r_{j'}^{(2)} = 0 \\ j' \neq j \\ p_{ij'} = 1}} x_{ij'}^0 = 0.$$

En particulier,  $x_{ij}^0 = 0$ . On modifie cette solution en posant  $x_{ij}^0 = 1$  et  $\forall k \in \mathcal{S} \setminus \{0\}, x_{ij}^k = 0$  (on change la date de rencontre de ce couple pour la planifier à la date zéro) de manière à obtenir une solution appartenant à  $\delta_0(\Omega)$ . Cette solution est valide car aucun des deux participants n'avait de rencontre à la date zéro dans la solution initiale. De plus, la nouvelle valeur de l'objectif est conservée car la date de la rencontre entre ces deux participants a été avancée et ne pénalise donc pas la fonction objectif.  $\square$

### 8.5.2 Rencontres aux dates $r_{ij}$ ( $\mathbf{D}r_{ij}$ )

On peut généraliser la règle de dominance à la date zéro pour chaque date  $t \in \{r_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$ .

**Règle de dominance 8.5.3.** Pour chaque couple  $(M_i, F_j) \in \mathcal{M} \times \mathcal{F}$  devant se rencontrer, une de leurs rencontres peut se dérouler à la date  $r_{ij}$ . On note  $\delta_t(\Omega)$ , l'ensemble des solutions vérifiant la contrainte (8.35) à la date  $t \in \mathcal{S}$ .

La règle de dominance 8.5.3 se traduit par les contraintes (8.35) pour le modèle indexé sur le temps.

$$\forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, \text{ t.q. } p_{ij} = 1 \text{ et } t = r_{ij}, \quad x_{ij}^t + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ p_{i'j} = 1}} x_{i'j}^t + \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ p_{ij'} = 1}} x_{ij'}^t \geq 1 \quad (8.35)$$

De même que pour la règle de dominance à la date zéro, il n'est pas intéressant d'écrire ces contraintes pour le modèle basé sur les dates des rencontres.

**Proposition 8.5.4.** L'ensemble  $\bigcap_{t \in \mathcal{S}} \delta_t(\Omega)$  est non vide et contient au moins une solution optimale.

*Démonstration.* Cette démonstration est illustrée sur l'exemple 8.5.2 ci-après.

On raisonne par récurrence sur  $t \in \mathcal{S}$ . D'après la proposition 8.5.2,  $\delta_0(\Omega)$  contient une solution optimale, ce qui initialise la récurrence. Soit  $t \in \mathcal{S}$ , on suppose que  $\bigcap_{t' \in \llbracket 0, t \rrbracket} \delta_{t'}(\Omega)$  contient une solution optimale. On démontre que

$\bigcap_{t' \in \llbracket 0, t+1 \rrbracket} \delta_{t'}(\Omega)$  contient une solution optimale. Soit une solution optimale appartenant à  $\bigcap_{t' \in \llbracket 0, t \rrbracket} \delta_{t'}(\Omega)$ . On modifie

séquentiellement cette instance pour obtenir une instance appartenant à  $\bigcap_{t' \in \llbracket 0, t+1 \rrbracket} \delta_{t'}(\Omega)$ . On détermine un ordre sur

les rencontres dont l'un des participants a sa date d'arrivée égale à  $t+1$  et supérieure ou égale à celle de sa/son partenaire. Pour chacune de ces rencontres prises dans l'ordre déterminé, si cette rencontre respecte la contrainte (8.35) correspondante, on ne modifie pas l'instance. Sinon, on planifie la date de rendez-vous à la date  $t+1$ . On obtient une instance valide car aucun des deux participants concernés n'avait de rencontre planifiée à la date  $t+1$  avant cette modification. De plus cette solution reste optimale car la nouvelle date de cette rencontre est antérieure à la précédente et n'augmente donc pas la valeur de l'objectif. En outre, il n'est pas possible que cela diminue la valeur de l'objectif car cela reviendrait à dire que la solution initiale n'était pas optimale. Enfin, la solution obtenue appartient toujours à l'ensemble  $\bigcap_{t' \in \llbracket 0, t \rrbracket} \delta_{t'}(\Omega)$ . En effet, la rencontre dont la date a été modifiée ne peut pas avoir lieu à une date inférieure

ou égale à  $t$  car l'un des participants arrive à la date  $t+1$ . L'ensemble  $\mathcal{S}$  étant fini,  $\bigcap_{t' \in \mathcal{S}} \delta_{t'}(\Omega)$  est donc non vide et contient au moins une solution optimale.  $\square$

**Exemple 8.5.2.** La figure 8.4 présente une solution optimale d'une instance de  $1-1MS|(U \cup V, E), r_i^{(1)}, r_j^{(2)}|W_{\max}$ . Cette solution ne vérifie pas les contraintes (8.35) en chacune des rencontres grisées sur la figure. Par exemple, pour



le couple  $(M_1, F_2)$ , on a  $t = \max(r_1^{(1)}, r_2^{(2)}) = 1$ , or l'homme  $M_1$  effectue ses rencontres aux dates 2, 3 et 5 et la femme  $F_2$  effectue ses rencontres aux dates 2, 3 et 4.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	4	4	$W_i^{(1)}$
$M_0$	0	4	1	2	5		2
$M_1$	1	2		3		5	2
$M_2$	3		3	4	6		1
$M_3$	3	3	4		7	6	1
	$W_j^{(2)}$	2	2	2	1	1	$W_{\max}^* = 2$

FIGURE 8.4 – Solution optimale non dominante.

On peut transformer cette instance pour qu'elle appartienne à  $\delta_0(\Omega)$  et reste optimale en modifiant la date de rencontre entre  $M_0$  et  $F_0$  comme indiqué dans la figure 8.5. On remarque que la rencontre entre  $M_0$  et  $F_3$  ne vérifie plus les contraintes (8.35) pour  $t = 4$ . Cette rencontre sera traitée par la suite.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	4	4	$W_i^{(1)}$
$M_0$	0	<b>0</b>	1	2	5		2
$M_1$	1	2		3		5	2
$M_2$	3		3	4	6		1
$M_3$	3	3	4		7	6	1
	$W_j^{(2)}$	1	2	2	1	1	$W_{\max}^* = 2$

FIGURE 8.5 – Solution optimale de  $\delta_0(\Omega)$ .

Modifier la date de la rencontre entre l'homme  $M_1$  et la femme  $F_0$  permet d'obtenir une solution optimale dans  $\delta_1(\Omega)$  qui appartient encore à  $\delta_0(\Omega)$  comme indiqué dans la figure 8.6.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_j^{(2)} \setminus r_i^{(1)}$	0	0	0	4	4	$W_i^{(1)}$
$M_0$	0	0	1	2	5		2
$M_1$	1	<b>1</b>		3		5	2
$M_2$	3		3	4	6		1
$M_3$	3	3	4		7	6	1
	$W_j^{(2)}$	1	2	2	1	1	$W_{\max}^* = 2$

FIGURE 8.6 – Solution optimale de  $\delta_0(\Omega) \cap \delta_1(\Omega)$ .

La solution de la figure 8.6 appartient également à  $\delta_3(\Omega)$  car les participants  $M_2, M_3, F_0, F_1$  et  $F_2$  ont tous une rencontre à effectuer à la date 3. On obtient facilement une solution optimale de  $\delta_4(\Omega)$  en modifiant les dates de rencontre des couples  $(M_0, F_3)$  et  $(M_1, F_4)$  comme présenté dans la figure 8.7.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_j^{(2)} \setminus r_i^{(1)}$	0	0	0	4	4	$W_i^{(1)}$
$M_0$	0	0	1	2	<b>4</b>		1
$M_1$	1	1		3		<b>4</b>	1
$M_2$	3		3	4	6		1
$M_3$	3	3	4		7	6	1
	$W_j^{(2)}$	1	2	2	1	1	$W_{\max}^* = 2$

FIGURE 8.7 – Solution optimale de  $\delta_0(\Omega) \cap \delta_1(\Omega) \cap \delta_3(\Omega) \cap \delta_4(\Omega)$ .



### 8.5.3 Ordonnancements actif et semi-actif (AS et SAS)

Dans les problèmes classiques d'ordonnancement, on dit qu'un ordonnancement est actif si aucune tâche ne peut être exécutée plus tôt sans retarder une autre tâche ou violer des contraintes. On dit qu'un ordonnancement est semi-actif si aucune tâche ne peut être exécutée plus tôt sans changer l'ordre des tâches ou violer des contraintes (voir (Brucker, 2001)).

**Règle de dominance 8.5.5.** On note respectivement  $\delta_a(\Omega)$  et  $\delta_{sa}(\Omega)$  les ensembles d'ordonnancement actifs et semi-actifs. Trivialement, l'ensemble des ordonnancements actifs est inclus dans l'ensemble des ordonnancements semi-actifs.

**Exemple 8.5.3.** La figure 8.8 présente une solution optimale semi-active mais non active pour une instance avec une rencontre interdite entre  $F_1$  et  $M_0$ . En effet, la rencontre entre la femme  $F_1$  et l'homme  $M_1$  peut avoir lieu à la date zéro, elle ne peut cependant pas avoir lieu plus tôt que la date trois sans modifier l'ordre des rencontres.

Participants\Créneaux	0	1	2	3	$W_i^{(1)}$
$M_0$	$F_2$	-	$F_0$		1
$M_1$	-	$F_0$	$F_2$	$F_1$	1
$M_2$	$F_0$	$F_1$	-	$F_2$	1
$M_3$		$F_2$	$F_1$	$F_0$	1
					$W_j^{(2)}$
$F_0$	$M_2$	$M_1$	$M_0$	$M_3$	0
$F_1$	-	$M_2$	$M_3$	$M_1$	1
$F_2$	$M_0$	$M_3$	$M_1$	$M_2$	0
			$W_{\max}^*$		1

FIGURE 8.8 – Ordonnancement semi-actif.

Les contraintes suivantes permettent d'obtenir respectivement un ordonnancement actif et semi-actif dans le modèle indexé sur le temps.

$$\forall M_i \in \mathcal{M}, F_j \in \mathcal{F} \text{ t.q. } p_{ij} = 1, \forall k \in \llbracket r_{ij} + 1, s - 1 \rrbracket, \forall k' \in \llbracket r_{ij}, k - 1 \rrbracket, x_{ij}^k \leq \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ r_{ij'}^{(2)} \leq k'}} x_{ij'}^{k'} + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ r_{i'}^{(1)} \leq k'}} x_{i'j}^{k'} \quad (8.36)$$

$$\forall M_i \in \mathcal{M}, F_j \in \mathcal{F} \text{ t.q. } p_{ij} = 1, \forall k \in \llbracket r_{ij} + 1, s - 1 \rrbracket, k' = k - 1, x_{ij}^k \leq \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ r_{ij'}^{(2)} \leq k'}} x_{ij'}^{k'} + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ r_{i'}^{(1)} \leq k'}} x_{i'j}^{k'} \quad (8.37)$$

Les contraintes (8.36) traduisent le fait que si la rencontre entre les participants  $M_i$  et  $F_j$  a lieu à la date  $k$ , alors pour toutes les dates précédant  $k$ , au moins l'un de ces deux participants effectue une rencontre à cette date. Les contraintes (8.37) sont moins fortes car elles ne s'appliquent qu'à la date précédant  $k$  immédiatement. Ces dernières assurent d'obtenir un ordonnancement semi-actif.

**Proposition 8.5.6.** Les ensembles  $\delta_a(\Omega)$  et  $\delta_{sa}(\Omega)$  contiennent au moins une solution optimale pour le critère  $W_{\max}$ .

*Démonstration.* Les ordonnancements actifs et semi-actifs contiennent une solution optimale pour les critères réguliers (Baker, 1974). Un critère est dit régulier s'il est fonction croissante des dates de fin des tâches. L'objectif  $W_{\max}$  est régulier. En effet, ordonnancer une rencontre plus tard n'améliore jamais le nombre d'attentes des participants.  $\square$

**Proposition 8.5.7.** L'ensemble  $\bigcap_{t \in S} \delta_t(\Omega)$  contient strictement l'ensemble des ordonnancements actifs  $\delta_a(\Omega)$ .

*Démonstration.*

$$1. \delta_a(\Omega) \subset \bigcap_{t \in \mathcal{S}} \delta_t(\Omega).$$

Soit une solution appartenant à  $\delta_a(\Omega)$ , démontrons que cette solution vérifie les contraintes (8.35), c'est-à-dire :

$$\forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket, \text{ t.q. } p_{ij} = 1 \text{ et } t = r_{ij}, \quad x_{ij}^t + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ p_{i'j}=1}} x_{i'j}^t + \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ p_{ij'}=1}} x_{ij'}^t \geq 1.$$

Soit un couple  $(F_i, M_j)$  devant se rencontrer. Si ce couple se rencontre à la date  $r_{ij}$ , alors  $x_{ij}^{r_{ij}} = 1$  et la contrainte est vérifiée. Dans le cas contraire, il existe  $k > r_{ij}$  tel que  $x_{ij}^k = 1$ . D'après les contraintes (8.36),

$$\forall k' \in \llbracket r_{ij}, k-1 \rrbracket, x_{ij}^k \leq \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ r_{j'}^{(2)} \leq k'}} x_{ij'}^{k'} + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ r_{i'}^{(1)} \leq k'}} x_{i'j}^{k'}.$$

En particulier, pour  $k' = r_{ij}$ , on obtient :

$$1 \leq \sum_{\substack{j' \in \llbracket 0, n-1 \rrbracket \\ j' \neq j \\ r_{j'}^{(2)} \leq k'}} x_{ij'}^{k'} + \sum_{\substack{i' \in \llbracket 0, m-1 \rrbracket \\ i' \neq i \\ r_{i'}^{(1)} \leq k'}} x_{i'j}^{k'}.$$

Ce qui prouve que les contraintes (8.35) sont respectées.

$$2. \delta_a(\Omega) \neq \bigcap_{t \in \mathcal{S}} \delta_t(\Omega).$$

On vérifie facilement que l'ordonnancement de la figure (8.9) est dans  $\bigcap_{t \in \mathcal{S}} \delta_t(\Omega)$ , cependant, il n'est pas actif car la rencontre entre l'homme  $M_0$  et la femme  $F_2$  peut être positionnée plus tôt à la date 2 au lieu de 3. (Cet ordonnancement n'est donc pas non plus semi-actif.)

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_j^{(2)} \setminus r_i^{(1)}$	0	0	0	1	$W_i^{(1)}$
$M_0$	0	1	0	3	4	1
$M_1$	0		3	1	2	1
$M_2$	0	2	1	0	3	0
	$W_j^{(2)}$	1	1	1	1	$W_{\max}^* = 1$

FIGURE 8.9 – Ordonnancement de  $\bigcap_{t \in \mathcal{S}} \delta_t(\Omega)$  non actif .

□

### 8.5.4 Symétries (Sym)

On note  $\mathcal{M}_{F_j} := \{M_i \in \mathcal{M} \mid p_{ij} = 1\}$  (où  $j \in \llbracket 0, n-1 \rrbracket$ ) l'ensemble des participants devant rencontrer la femme  $F_j$ . De même, on note  $\mathcal{F}_{M_i} := \{F_j \in \mathcal{F} \mid p_{ij} = 1\}$  (où  $i \in \llbracket 0, m-1 \rrbracket$ ) l'ensemble des participants devant rencontrer l'homme  $M_i$ .

Deux participants ayant la même date d'arrivée et la même liste de rencontres peuvent échanger leur rôle. On peut casser ces symétries à l'aide des contraintes (8.38) et (8.39).

$$\forall i_1, i_2 \in \llbracket 0, m-1 \rrbracket \text{ t.q. } r_{i_1}^{(1)} = r_{i_2}^{(1)}, i_1 < i_2, \mathcal{F}_{M_{i_1}} = \mathcal{F}_{M_{i_2}}, j_0 = \min\{j \mid F_j \in \mathcal{F}_{M_{i_1}}\}, \quad C_{i_1 j_0} < C_{i_2 j_0} \quad (8.38)$$

$$\forall j_1, j_2 \in \llbracket 0, n-1 \rrbracket \text{ t.q. } r_{j_1}^{(2)} = r_{j_2}^{(2)}, j_1 < j_2, \mathcal{M}_{F_{j_1}} = \mathcal{M}_{F_{j_2}}, i_0 = \min\{i \mid M_i \in \mathcal{M}_{F_{j_1}}\}, \quad C_{i_0 j_1} < C_{i_0 j_2} \quad (8.39)$$

Ces contraintes permettent de casser les symétries entre deux participants ayant la même date d'arrivée et la même liste de rencontres à effectuer. Pour deux participants équivalents de ce point de vue, on force l'un des deux à effectuer la première rencontre de sa liste avant l'autre.

Ces contraintes peuvent être réécrites de la manière suivante pour le modèle indexé sur le temps :

$$\begin{aligned} \forall i_1, i_2 \in \llbracket 0, m-1 \rrbracket \text{ t.q. } r_{i_1}^{(1)} = r_{i_2}^{(1)}, i_1 < i_2, \mathcal{F}_{M_{i_1}} = \mathcal{F}_{M_{i_2}}, j_0 = \min\{j \mid F_j \in \mathcal{F}_{M_{i_1}}\}, \forall k' \in \llbracket r_{i_1 j_0}, s-1 \rrbracket, \\ \sum_{k=r_{i_1 j_0}}^{k'} x_{i_1 j_0}^k \geq \sum_{k=r_{i_1 j_0}}^{k'} x_{i_2 j_0}^k \end{aligned} \quad (8.40)$$

$$\begin{aligned} \forall j_1, j_2 \in \llbracket 0, n-1 \rrbracket \text{ t.q. } r_{j_1}^{(2)} = r_{j_2}^{(2)}, j_1 < j_2, \mathcal{M}_{F_{j_1}} = \mathcal{M}_{F_{j_2}}, i_0 = \min\{i \mid M_i \in \mathcal{M}_{F_{j_1}}\}, \forall k' \in \llbracket r_{i_0 j_1}, s-1 \rrbracket, \\ \sum_{k=r_{i_0 j_1}}^{k'} x_{i_0 j_1}^k \geq \sum_{k=r_{i_0 j_1}}^{k'} x_{i_0 j_2}^k. \end{aligned} \quad (8.41)$$

Les contraintes de symétrie s'écrivent de la manière suivante pour le modèle basé sur les dates de rencontres.

$$\forall i_1, i_2 \in \llbracket 0, m-1 \rrbracket \text{ t.q. } r_{i_1}^{(1)} = r_{i_2}^{(1)}, i_1 < i_2, \mathcal{F}_{M_{i_1}} = \mathcal{F}_{M_{i_2}}, j_0 = \min\{j \mid M_j \in \mathcal{F}_{M_{i_1}}\}, \quad \delta_{i_1 i_2}^{j_0} = 1 \quad (8.42)$$

$$\forall j_1, j_2 \in \llbracket 0, n-1 \rrbracket \text{ t.q. } r_{j_1}^{(2)} = r_{j_2}^{(2)}, j_1 < j_2, \mathcal{M}_{F_{j_1}} = \mathcal{M}_{F_{j_2}}, i_0 = \min\{i \mid F_i \in \mathcal{M}_{F_{j_1}}\}, \quad \eta_{j_1 j_2}^{i_0} = 1 \quad (8.43)$$

**Règle de dominance 8.5.8.** On note  $\delta_f(\Omega)$  l'ensemble des solutions vérifiant les contraintes (8.38) et  $\delta_m(\Omega)$  l'ensemble des solutions vérifiant les contraintes (8.39).

**Exemple 8.5.4.** La figure 8.10 représente une solution optimale pour une instance avec retards et rencontres interdites. On remarque que dans cette instance, le rôle des femmes  $F_0$  et  $F_1$  est symétrique car elles arrivent à la même date et ont la même liste d'hommes à rencontrer. De même, le rôle des femmes  $F_2$  et  $F_3$  est symétrique. Les hommes  $M_0$ ,  $M_1$  et  $M_2$  ont également la même date d'arrivée et la même liste de rencontres, et de même pour les hommes  $M_3$  et  $M_4$ . Ainsi, on peut permuter le rôle de chacun des participants dans l'un des ensembles suivants :  $\{M_0, M_1, M_2\}$ ,  $\{M_3, M_4\}$ ,  $\{F_0, F_1\}$  ou  $\{F_2, F_3\}$ . Dans cette solution, les contraintes (8.38) et (8.39) ne sont pas respectées. En effet, la femme  $F_0$  ne rencontre pas les hommes  $M_0$ ,  $M_1$  et  $M_2$  dans cet ordre et l'homme  $M_0$  ne rencontre pas les femmes  $F_0$ ,  $F_1$ ,  $F_2$  et  $F_3$  dans cet ordre.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	0	1	$W_i^{(1)}$
$M_0$	0	2	1	4	0	3	0
$M_1$	0	1	2	0	4	5	1
$M_2$	0	0	3	2	1	4	0
$M_3$	1			1	3	2	0
$M_4$	1			3	2	1	0
	$W_j^{(2)}$	0	1	0	0	0	$W_{\max}^* = 1$

FIGURE 8.10 – Solution optimale avec  $W_{\max}^* = 1$ .

On peut obtenir une solution de l'ensemble dominant  $\delta_m(\Omega)$  en permutant certaines lignes de ce tableau. On peut permuter deux lignes si les participants impliqués ont la même date d'arrivée et la même liste de rencontres sans

modifier la valeur de l'objectif. En intervertissant le rôle des hommes  $M_0$  et  $M_2$ , on obtient une solution de  $\delta_m(\Omega)$  présentée dans la figure 8.11. Cette solution vérifie les contraintes (8.38) car les parties grisées sont ordonnées.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	0	1	$W_i^{(1)}$
$M_0$	0	0	3	2	1	4	0
$M_1$	0	1	2	0	4	5	1
$M_2$	0	2	1	4	0	3	0
$M_3$	1			1	3	2	0
$M_4$	1			3	2	1	0
	$W_j^{(2)}$	0	1	0	0	0	$W_{\max}^* = 1$

FIGURE 8.11 – Solution optimale dominante pour les femmes.

À partir de la première solution présentée dans la figure (8.10), on peut déterminer une solution respectant les contraintes (8.39) en intervertissant le rôle de la femme  $F_0$  avec la femme  $F_1$  et en intervertissant le rôle de la femme  $F_2$  avec la femme  $F_3$ . On obtient la solution présentée dans la figure (8.12). Cette solution appartient à  $\delta_f(\Omega)$  car les parties grisées sont ordonnées.

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	0	1	$W_i^{(1)}$
$M_0$	0	1	2	0	4	3	0
$M_1$	0	2	1	4	0	5	1
$M_2$	0	3	0	1	2	4	0
$M_3$	1			3	1	2	0
$M_4$	1			2	3	1	0
	$W_j^{(2)}$	0	1	0	0	0	$W_{\max}^* = 1$

FIGURE 8.12 – Solution optimale dominante pour les hommes

À partir de cette dernière solution on peut obtenir une solution de  $\delta_f(\Omega) \cap \delta_m(\Omega)$  en intervertissant les rôles des hommes  $M_3$  et  $M_4$  comme présenté dans la figure (8.13). Dans cette solution, chacune des parties grisées est ordonnée, ce qui permet de vérifier les contraintes (8.38) et (8.39).

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	
	$r_j^{(2)} \setminus r_i^{(1)}$	0	0	0	0	1	$W_i^{(1)}$
$M_0$	0	1	2	0	4	3	0
$M_1$	0	2	1	4	0	5	1
$M_2$	0	3	0	1	2	4	0
$M_3$	1			2	3	1	0
$M_4$	1			3	1	2	0
	$W_j^{(2)}$	0	1	0	0	0	$W_{\max}^* = 1$

FIGURE 8.13 – Solution optimale dominante.

### Proposition 8.5.9.

1. Les ensembles  $\delta_f(\Omega)$  et  $\delta_m(\Omega)$  contiennent chacun au moins une solution optimale.
2. Conjecture :  $\delta_f(\Omega) \cap \delta_m(\Omega)$  contient au moins une solution optimale.

*Démonstration.*

1. Soit une instance et une solution optimale pour cette instance. On suppose que cette solution ne vérifie pas les contraintes (8.38). Cela signifie qu'il existe deux hommes  $M_{i_1}$  et  $M_{i_2}$  ( $i_1 \leq i_2$ ) ayant la même date d'arrivée et la même liste de rencontres  $\mathcal{F}_{M_{i_1}}$  telles que :  $C_{i_1 j_0} > C_{i_2 j_0}$  où  $F_{j_0}$  est la première femme de la liste  $\mathcal{F}_{M_{i_1}}$ . On peut transformer cette solution en intervertissant le rôle des deux hommes  $M_{i_1}$  et  $M_{i_2}$  sans changer la valeur de l'objectif car ces deux hommes ont exactement la même date d'arrivée et la même liste de femmes à rencontrer. On obtient alors une solution optimale appartenant à  $\delta_m(\Omega)$ . De même, on montre que  $\delta_f(\Omega)$  contient au moins une solution optimale en intervertissant le rôle des femmes dans l'énoncé précédent.

□

### 8.5.5 Interdiction de créneaux vides (NES) (No Empty Slot)

On peut interdire les créneaux vides à partir de la date  $r_{\max} := \max \{r_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\}$  et jusqu'à la fin des rencontres. En effet, si un tel créneau est vide, on peut décaler toutes les rencontres suivantes au créneau précédent. Cette contrainte peut s'écrire de la manière suivante pour le modèle indexé sur le temps :

$$\forall i' \in \llbracket 0, m-1 \rrbracket, j' \in \llbracket 0, n-1 \rrbracket, \text{ tels que } p_{i'j'} = 1, \forall k \in \llbracket r_{\max}+1, s-1 \rrbracket, \quad x_{i'j'}^k \leq \sum_{\substack{i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket \\ p_{ij} = 1}} x_{ij}^{k-1}. \quad (8.44)$$

Si le couple  $(M_{i'}, F_{j'})$  se rencontre à la date  $k$ , alors il y a au moins une rencontre qui est faite à la date  $k-1$ .

**Règle de dominance 8.5.10.** On note  $\delta_{nes}(\Omega)$  (no empty slot) l'ensemble des solutions n'ayant pas de créneaux vides, c'est-à-dire vérifiant les contraintes (8.44).

**Exemple 8.5.5.** La figure 8.14 représente une solution optimale d'une instance à deux femmes et quatre hommes.

		$F_0$	$F_1$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	3	$W_i^{(1)}$
$M_0$	0	0		0
$M_1$	0	1		1
$M_2$	0	2		<b>2</b>
$M_3$	3	3	5	1
	$W_j^{(2)}$	0	2	$W_{\max}^* = 2$

FIGURE 8.14 – Solution optimale non dominante

Dans cet exemple,  $r_{\max} = 3$ , il s'agit de la première date à laquelle tous les participants sont arrivés. Dans cette solution, aucune rencontre n'a lieu à la date 4. On peut évidemment changer la date de la rencontre entre  $M_3$  et  $F_1$  à la date 4 sans dégrader l'objectif ainsi que le montre la solution présentée dans la figure 8.15.

		$F_0$	$F_1$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	3	$W_i^{(1)}$
$M_0$	0	0		0
$M_1$	0	1		1
$M_2$	0	2		<b>2</b>
$M_3$	3	3	4	0
	$W_j^{(2)}$	0	1	$W_{\max}^* = 2$

FIGURE 8.15 – Solution optimale dominante

**Proposition 8.5.11.** L'ensemble  $\delta_{nes}(\Omega)$  contient au moins une solution optimale.

*Démonstration.* Considérons une solution optimale ne vérifiant pas les contraintes (8.44). Soit  $k$  la plus petite date supérieure ou égale à  $r_{\max}$  à laquelle aucune rencontre n'a lieu. Il existe au moins une date strictement supérieure à  $k$  notée  $k'$  à laquelle ont lieu des rencontres. Toutes les rencontres ayant lieu à partir de la date  $k'$  sont réordonnées  $k' - k$  créneaux plus tôt. On procède ainsi de manière itérative jusqu'à ce qu'aucun créneau ne soit vide après la date  $r_{\max}$ .  $\square$

## 8.6 Résultats expérimentaux

Dans cette section, nous présentons les résultats numériques obtenus pour les différents modèles de ce chapitre. Nous nous intéressons à la relaxation continue de ces modèles ainsi qu'aux temps de résolution sur l'ensemble des 300 instances générées. Pour ces tests, nous avons utilisé le solveur ILOG Cplex (version 12.5) sur un ordinateur sous windows server 2008 avec un processeur Intel Core I3-540 et 8 Go de RAM. Les tableaux 8.1 et 8.2 rappellent les configurations qui nous intéressent et présentent les notations utilisées.

Notation	Contraintes :
Modèle de base indexé sur le temps	
Ag (Agrégé)	$\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket$ tels que $p_{ij} = 1$ , $\sum_{k=\max(r_{ij}, d_{ij}+LB)}^{\min(d_{ij}+UB-1, s-1)} (k+1)x_{ij}^k - d_{ij} \leq W_{\max}$
KS (Knapsack)	$\forall k \in \llbracket LB, UB-1 \rrbracket, (k+1)x_{ij}^{e_{ij}+k} \leq W_{\max}$
IV	$\left\{ \begin{array}{l} \forall i \in \llbracket 0, m-1 \rrbracket, W_{\max} \geq e_i^{(1)} + LB - \sum_{k=r_i^{(1)}}^{e_i^{(1)}+LB-1} \sum_{\substack{j \in \llbracket 0, n-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k - r_i^{(1)} \\ \forall j \in \llbracket 0, n-1 \rrbracket, W_{\max} \geq e_j^{(2)} + LB - \sum_{k=r_j^{(2)}}^{e_j^{(2)}+LB-1} \sum_{\substack{i \in \llbracket 0, m-1 \rrbracket \\ p_{ij}=1}} x_{ij}^k - r_j^{(2)} \end{array} \right.$

Tableau 8.1 – Configurations de test

Modèle indexé sur le temps étendu (E)	
IVe	$\left\{ \begin{array}{l} \forall k \in \mathcal{S}, \text{ si } nbF_k < nbM'_k, \sum_{i \in \llbracket 0, m-1 \rrbracket} \sum_{k=r_i^{(1)}}^{e_i^{(1)}-1} x_{iw}^k \geq nbM'_k - nbF_k \\ \forall k \in \mathcal{S}, \text{ si } nbM_k < nbF'_k, \sum_{j \in \llbracket 0, n-1 \rrbracket} \sum_{k=r_j^{(2)}}^{e_j^{(2)}-1} x_{wj}^k \geq nbF'_k - nbM_k. \end{array} \right.$
Modèle basé sur les dates de rencontres ( $C_{ij}$ )	
IVt	$\left\{ \begin{array}{l} \forall j \in \llbracket 0, n-1 \rrbracket, i_1, i_2, i_3 \in \llbracket 0, m-1 \rrbracket, \text{ t.q. } i_1 < i_2 < i_3, p_{i_1j} = p_{i_2j} = p_{i_3j} = 1, \\ 1 \leq \delta_{i_1i_2}^j + \delta_{i_2i_3}^j + \delta_{i_3i_1}^j \leq 2 \\ \forall i \in \llbracket 0, m-1 \rrbracket, j_1, j_2, j_3 \in \llbracket 0, n-1 \rrbracket, \text{ t.q. } i_1 < i_2 < i_3, p_{ij_1} = p_{ij_2} = p_{ij_3} = 1, \\ 1 \leq \eta_{j_1j_2}^i + \eta_{j_2j_3}^i + \eta_{j_3j_1}^i \leq 2 \end{array} \right.$

Tableau 8.2 – Configurations de test (suite)

### 8.6.1 Relaxation continue

Dans un premier temps, on s'intéresse aux bornes inférieures déterminées par relaxation continue des différents modèles.

Le tableau 8.3 présente les moyennes des différences relatives pour les bornes inférieures obtenues par relaxation continue en fonction de la configuration, du nombre de participants et de la proportion entre hommes et femmes. (Le calcul de la différence relative est  $(Opt - lb)/Opt$ ). On dispose de toutes les valeurs optimales des objectifs (résolus par la PLNE), on peut donc présenter les différences relatives à l'optimum. La première colonne correspond au modèle indexé sur le temps, la deuxième au le modèle indexé sur le temps étendu, la troisième au modèle basé sur les dates de rencontres. À titre de comparaison, on ajoute dans la dernière colonne les différences relatives moyennes entre la valeur optimale de l'objectif et la meilleure borne inférieure calculée a priori présentée dans le chapitre 6). Les inégalités valides apportent une bonne amélioration de la borne inférieure par rapport au modèle de base (Ag) dont la relaxation est très mauvaise. Cependant, les bornes inférieures calculées a priori dominent largement l'ensemble des bornes trouvées par relaxation des modèles.

	Ag	KS	AgIV	KSIV	EAg	EKS	EAgIV	EKSIV	EAgIVe	EKSIVe	EAgIVVe	EKSIVVe	$C_{ijIVt}$	$LB_{best}$
14 p.														
40%/60%	90%	60%	<b>41%</b>	45%	47%	44%	<b>30%</b>	<b>30%</b>	31%	31%	<b>30%</b>	<b>30%</b>	95%	<b>15%</b>
50%/50%	94%	65%	81%	<b>64%</b>	48%	42%	<b>38%</b>	39%	41%	41%	<b>38%</b>	39%	97%	<b>2%</b>
60%/40%	93%	59%	52%	<b>51%</b>	48%	45%	30%	32%	30%	33%	<b>29%</b>	31%	97%	<b>12%</b>
24 p.														
40%/60%	100%	72%	48%	<b>47%</b>	62%	55%	29%	34%	31%	35%	<b>27%</b>	32%	100%	<b>15%</b>
50%/50%	100%	90%	100%	<b>87%</b>	58%	52%	<b>42%</b>	44%	48%	46%	<b>42%</b>	44%	100%	<b>2%</b>
60%/40%	100%	85%	66%	<b>65%</b>	58%	55%	27%	32%	27%	32%	<b>26%</b>	30%	100%	<b>7%</b>
40 p.														
40%/60%	100%	94%	<b>52%</b>	<b>52%</b>	89%	73%	40%	41%	40%	41%	<b>38%</b>	39%	100%	<b>8%</b>
50%/50%	100%	93%	100%	<b>90%</b>	53%	48%	<b>39%</b>	40%	45%	44%	<b>39%</b>	40%	100%	<b>8%</b>
60%/40%	100%	94%	<b>59%</b>	<b>59%</b>	77%	72%	34%	40%	34%	40%	<b>32%</b>	38%	100%	<b>7%</b>
50 p.														
40%/60%	100%	95%	<b>51%</b>	<b>51%</b>	79%	73%	35%	41%	35%	41%	<b>34%</b>	39%	100%	<b>6%</b>
50%/50%	100%	94%	100%	<b>91%</b>	83%	69%	65%	<b>60%</b>	72%	64%	65%	<b>60%</b>	100%	<b>5%</b>
60%/40%	100%	93%	<b>58%</b>	<b>58%</b>	79%	65%	35%	37%	35%	37%	<b>34%</b>	36%	100%	<b>2%</b>
70 p.														
40%/60%	100%	96%	<b>47%</b>	<b>47%</b>	99%	83%	40%	40%	40%	40%	<b>39%</b>	<b>39%</b>	100%	<b>4%</b>
50%/50%	100%	96%	100%	<b>93%</b>	77%	62%	58%	<b>51%</b>	68%	57%	58%	<b>51%</b>	100%	<b>2%</b>
60%/40%	100%	97%	<b>49%</b>	<b>49%</b>	89%	77%	34%	34%	34%	34%	<b>33%</b>	<b>33%</b>	100%	<b>4%</b>

Tableau 8.3 – Différences relatives moyennes entre les optima et les bornes inférieures de la relaxation continue

## 8.6.2 Temps de résolution

Dans cette section, nous nous intéressons à la proportion d'instances résolues à l'optimum en moins d'une demi-heure ainsi qu'aux temps moyens de résolution pour les différents modèles de PLNE développés dans ce chapitre.

### Sans bornes

Le tableau 8.4 présente les proportions d'instances résolues à l'optimum ainsi que les temps de résolution moyens pour les configurations de base sans bornes (la moyenne des temps est systématiquement faite sur les instances résolues en moins d'une demi-heure). Pour ces tests, on fixe un temps limite de calcul d'une demi-heure. De manière générale, l'ordre de grandeur des temps de résolution est de la minute pour les instances de taille 40, entre 5 et 10 min pour les instances de taille 50. Très peu d'instances à 70 participants sont résolues dans le temps limite (30 min). On constate que les configurations apportant les meilleures bornes inférieures ne sont pas celles qui procurent les meilleurs résultats en terme de temps de calcul, en particulier, ajouter les inégalités valides n'apporte pas d'amélioration significative.

Configuration :	Ag	KS	AgIV	KSIV
14 p.	60/60 <b>0,07s</b>	60/60 0,11s	60/60 0,09s	60/60 0,12s
24 p.	60/60 <b>0,94s</b>	60/60 1,47s	60/60 1,05s	60/60 1,44s
40 p.	60/60 <b>40,96s</b>	60/60 66,30s	60/60 64,56s	60/60 61,75s
50 p.	60/60 <b>255,70s</b>	60/60 520,85s	59/60 296,03s	60/60 353,16s
70 p.	<b>6/60</b> <b>899,50s</b>	1/60 1616,40s	3/60 1192,20s	<b>6/60</b> 1140,70s

Tableau 8.4 – Proportions d’instances résolues à l’optimum et temps de résolution moyens pour le modèle de base sans bornes

### Avec bornes

On obtient les résultats du tableau 8.5 en fournissant au modèle les meilleures bornes inférieures et supérieures au modèle (cf. les chapitres 6 et 7). Le temps limite de calcul est toujours fixé à la demi-heure. Les instances de taille 40 sont résolues en environ 3s et celles de taille 50 en environ 13s. Les différentes configurations procurent des résultats proches pour ces instances. En ce qui concerne les instances de taille 70, on améliore de beaucoup la proportion d’instances résolues à l’optimum par rapport aux configurations sans bornes. Plus de 90% sont résolues à l’optimum pour la configuration Ag (contraintes agrégées). De la même manière que pour les instances de taille plus petite, les ordres de grandeur de temps de résolution sont similaires selon les différentes configurations.

Configuration :	Ag	KS	AgIV	KSIV
14 p.	60/60 0,01s	60/60 0,01s	60/60 0,01s	60/60 0,01s
24 p.	60/60 0,14s	60/60 0,13s	60/60 0,14s	60/60 0,14s
40 p.	60/60 2,83s	60/60 <b>2,50s</b>	60/60 3,51s	60/60 3,28s
50 p.	60/60 13,99s	60/60 12,92s	60/60 <b>11,69s</b>	60/60 12,22s
70 p.	<b>55/60</b> 226,11s	54/60 202,69s	54/60 199,97s	54/60 212,17s

Tableau 8.5 – Proportions d’instances résolues à l’optimum et temps de résolution moyens pour le modèle de base avec bornes

### Avec bornes et solution de départ

Dans le tableau 8.6, on présente les résultats obtenus si on fournit au solveur une solution de départ. Cette solution est la meilleure obtenue par l’une des heuristiques du chapitre 7. Les résultats sont nettement améliorés pour les instances de taille 70. Pour la configuration Ag, on obtient trois solutions optimales de plus pour les instances à 70 participants.

Lorsqu’on lance la résolution sans limite de temps, les deux instances non résolues en moins d’une demi heure pour la configuration Ag sont résolues en respectivement 40 min et 1h07min.

Le tableau 8.7 présente les résultats obtenus dans les mêmes conditions en fonction de la proportion femmes/hommes. Ce tableau illustre la difficulté à résoudre les instances carrées. On remarque en particulier que l’ensemble des instances non résolues en moins d’une demi-heure sont des instances carrées quelle que soit la configuration.



Configuration :	Ag	KS	AgIV	KSIV
14 p.	60/60 <b>0,01s</b>	60/60 0,02s	60/60 <b>0,01s</b>	60/60 <b>0,01s</b>
24 p.	60/60 0,11s	60/60 0,11s	60/60 <b>0,10s</b>	60/60 <b>0,10s</b>
40 p.	60/60 2,05s	60/60 2,58s	60/60 <b>1,61s</b>	60/60 2,74s
50 p.	60/60 8,64s	60/60 10,58s	60/60 9,22s	60/60 <b>8,23s</b>
70 p.	<b>58/60</b> <b>284,79s</b>	54/60 341,63s	57/60 286,02s	53/60 358,74s

Tableau 8.6 – Proportions d’instances résolues à l’optimum et temps de résolution moyens pour le modèle de base avec bornes et solution de départ

Configuration :	Ag	KS	AgIV	KSIV
40%/60%	100/100 16,61s	100/100 22,85s	100/100 16,77s	100/100 <b>16,17s</b>
50%/50%	98/100 <b>139,77s</b>	94/100 156,35s	97/100 142,93s	93/100 160,70s
60%/40%	100/100 18,09s	100/100 22,63s	100/100 <b>14,18s</b>	100/100 31,17s

Tableau 8.7 – Proportions d’instances résolues à l’optimum et temps de résolution moyens pour le modèle de base avec bornes et solution de départ

### Modèle indexé sur le temps étendu (avec bornes et solution de départ)

Nous avons testé le modèle indexé sur le temps avec variables additionnelles en lui procurant les meilleures bornes inférieures avec solution de départ calculée a priori. Nous ne présentons que les temps de résolution des meilleures configurations dans le tableau 8.8. Nous constatons que malgré une meilleure relaxation continue, ces modèles procurent de moins bons résultats que le modèle de base.

	EAg	EAgIVe	EAgIVIVe
14 p.	60/60 <b>0,03s</b>	60/60 <b>0,03s</b>	60/60 <b>0,03s</b>
24 p.	60/60 0,30s	60/60 0,28s	60/60 <b>0,23s</b>
40 p.	60/60 <b>8,61s</b>	60/60 9,12s	60/60 9,01s
50 p.	60/60 <b>52,52s</b>	60/60 62,66s	60/60 58,87s
70 p.	38/60 <b>838,45s</b>	36/60 800,88s	31/60 759,62s

Tableau 8.8 – Proportions d’instances résolues à l’optimum et temps moyens de résolution pour le modèle de base étendu

### Modèle basé sur les dates de rencontres (avec bornes et solution de départ)

Des tests préliminaires ont montré que les modèles basés sur les dates de rencontres fournissent de très mauvais résultats même avec ajout d’inégalités valides. À titre d’exemple, nous montrons les résultats obtenus pour les instances à 14 participants dans le tableau 8.9.

	$C_{ij}$	$C_{ij}IVt$
14p.	52/60	51/60
	62,73s	41,51s

Tableau 8.9 – Proportions d’instances résolues à l’optimum et temps moyens de résolution pour le modèle basé sur les date de rencontres

### Règles de dominance (avec bornes et solution de départ)

Dans cette section, on s’intéresse aux résultats obtenus après ajout de règles de dominance. Le tableau 8.10 récapitule ces règles.

Notation	espace de recherche	description
$Dr_{ij}$	$\bigcap_{t \in \mathcal{S}} \delta_t(\Omega)$	contraintes de dates
SAS	$\delta_{sa}(\Omega)$	ordonnancements semi-actifs
AS	$\delta_a(\Omega)$	ordonnancements actifs
Sym	$\delta_f(\Omega) \cap \delta_m(\Omega)$	contraintes de symétries par rapport aux participants
NES	$\delta_{nes}(\Omega)$	ordonnancements sans créneaux vides

Tableau 8.10 – Récapitulatif des notations des règles de dominance

Dans le tableau 8.11, nous présentons les résultats obtenus pour la meilleure configuration du modèle de base (Ag) auquel on ajoute les règles de dominance présentées dans la section 8.5. Nous constatons que les temps de calcul ne sont pas améliorés, ils sont même largement dégradés par l’ajout des contraintes imposant que l’ordonnement soit actif. Il est possible que l’espace des solutions optimales devienne trop restreint après l’ajout de ces contraintes et que la recherche d’une solution optimale devienne plus difficile, malgré la diminution de l’espace des solutions total.

	$Dr_{ij}$	SAS	AS	Sym	NES
14p.	60/60 0,02s	60/60 0,03s	60/60 0,14s	60/60 0,02s	60/60 0,04s
24p.	60/60 0,16s	60/60 0,80s	60/60 9,30s	60/60 0,20s	60/60 0,95s
40p.	60/60 2,66s	60/60 38,18s	43/60 272,75s	60/60 4,31s	60/60 47,29s
50p.	60/60 12,09s	55/60 141,24s	18/60 55,20s	60/60 18,02s	60/60 291,03s

Tableau 8.11 – Proportions d’instances résolues à l’optimum et temps moyens de résolution avec règles de dominance

## 8.7 Considérations sur la génération de colonnes

Comme nous l’avons vu dans ce chapitre, la relaxation continue des modèles de programmation linéaire en nombres entiers ne procure pas de borne inférieure satisfaisante. Une manière d’obtenir de meilleures bornes inférieures consiste à réécrire le modèle de manière différente à l’aide de la décomposition de Dantzig-Wolfe (voir (Lasdon, 1970) ou (Minoux, 2007)). Cette décomposition fournit un programme linéaire appelé programme maître qui est équivalent au programme de départ mais dont la relaxation continue est meilleure. Le nombre de variables (appelées colonnes) du programme maître est tellement élevé qu’on ne les explicite pas toutes. La résolution du problème maître relaxé se fait par génération de colonnes. Dans un premier temps, le programme maître restreint est résolu de manière exacte avec quelques colonnes. Un sous-problème permet de générer des colonnes prometteuses qu’on ajoute au programme maître restreint au fur et à mesure de la résolution.

Dans notre cas, le programme linéaire en nombres entiers indexé sur le temps semble prometteur pour effectuer une décomposition de Dantzig-Wolfe. Deux choix naturels sont envisageables. Dans le premier, les colonnes du pro-

gramme maître représentent les rencontres qui ont lieu à une date  $d$  fixée de la soirée. Le sous-problème est donc un problème d'affectation. Dans la seconde option que nous avons envisagée, les colonnes représentent le planning de chacune des femmes (ou de chacun des hommes). Le sous-problème consiste donc à affecter des dates aux rencontres d'une femme. Il s'agit également d'un problème d'affectation. Les modèles de génération de colonnes sont présentés en annexe A. Dans les deux cas, on vérifie que la matrice de contraintes du sous-problème est totalement unimodulaire, c'est à dire que chacun des sous-déterminants de la matrice vaut -1, 0 ou 1. Si une matrice de contraintes d'un programme linéaire en nombres entiers est totalement unimodulaire alors une solution optimale trouvée par la relaxation continue de ce programme est entière et fournit une solution optimale du programme linéaire en nombres entiers (voir (Nemhauser et Wolsey, 1988) pour une démonstration de ce résultat). De ce fait, dans les deux cas, la relaxation continue du programme maître est équivalente à la relaxation continue du modèle de base et ne fournit donc pas de meilleure borne inférieure. C'est pour cela que nous avons abandonné la génération de colonnes pour notre problème.

## 8.8 Conclusion

Dans ce chapitre, nous avons présenté plusieurs modèles de programmation linéaire en nombres entiers pour le problème général de speed-dating. Dans le premier modèle, les variables sont indexées sur le temps, celui-ci peut-être étendu en ajoutant des variables représentant des participants fictifs. Un autre modèle dont les variables représentent les dates de rencontres a été également proposé. Plusieurs inégalités valides ont été proposées pour ces différents modèles améliorant la qualité de la relaxation continue de ceux-ci. Le modèle le plus performant s'est avéré être le modèle de base sans ajout d'inégalités valides. Notons que les apports des bornes inférieures et solutions développées dans les chapitres 6 et 7 ont permis d'améliorer grandement la performance de ces modèles. Dans un deuxième temps, nous avons proposé plusieurs règles de dominance dans le but de diminuer l'espace de recherche des solutions. Si l'ajout de ces règles de dominance n'a pas permis d'améliorer la performance des modèles présentés ici, elles restent intéressantes du point de vue de l'étude des instances.

Dans la suite nous développerons d'autres méthodes de résolution dans le but de réussir à traiter de manière plus efficace les plus grandes instances. Dans ces méthodes, nous ferons largement usage des règles de dominance décrites dans ce chapitre.

*Les modèles linéaires et règles de dominance de ce chapitre ont été présentés lors du Workshop MAPSP 2013 (Le Roux et al., 2013a) et de la conférence ROADEF 13 (Le Roux et al., 2013b).*

## Chapitre 9

# Programmation par contraintes

### Sommaire

<b>9.1</b>	<b>Introduction</b>	<b>99</b>
<b>9.2</b>	<b>Modèle et stratégies de recherche</b>	<b>100</b>
<b>9.3</b>	<b>Réduction du domaine de recherche</b>	<b>101</b>
<b>9.4</b>	<b>Résultats expérimentaux</b>	<b>103</b>
<b>9.5</b>	<b>Conclusion</b>	<b>105</b>

### 9.1 Introduction

La programmation par contraintes est une approche générique qui permet de modéliser un problème et de le résoudre par une approche exacte. Un modèle de programmation par contraintes est défini par un ensemble de variables qui possèdent chacune un domaine de valeurs et un ensemble de contraintes logiques liant les variables. Un avantage de la programmation par contraintes est que celle-ci offre une grande souplesse dans l'écriture des contraintes. La résolution d'un modèle de programmation par contraintes combine l'exploration d'un arbre de recherche et des algorithmes de filtrage, ces derniers servant à réduire les domaines des variables au cours de la recherche en détectant des infaisabilités vis-à-vis des contraintes. La programmation par contraintes est une méthode de résolution exacte offrant une bonne complémentarité à la programmation linéaire en nombres entiers car les techniques de résolution employées sont complètement différentes : la PLNE exploite des propriétés mathématiques afférentes à la linéarité des contraintes alors que l'approche de résolution de la PPC est basée sur les liens logiques entre les variables et fonctionne par détection d'infaisabilités. Des approches utilisant la PPC ont été développées avec succès pour la résolution de problèmes proches du problème de speed-dating. On peut citer le problème d'open shop traité dans (Malapert et al., 2012) dans lequel les auteurs combinent des stratégies de redémarrage avec des techniques de propagation et des heuristiques d'ordonnancement. Également, des approches utilisant la PPC sont étudiées pour des problèmes d'emploi du temps. Dans (Yoshikawa et al., 1996), les auteurs utilisent un modèle de PPC pour trouver une première solution qui est ensuite améliorée à l'aide d'une recherche locale. Dans (Valoux et Housos, 2003), les auteurs développent un modèle de PPC amélioré par des recherches locales et des calculs de bornes inférieures au cours de la recherche.

Dans ce chapitre, nous développons et testons un modèle de programmation par contraintes pour le problème de speed-dating. Nous verrons que ce problème s'exprime de façon très simple et ne nécessite que des contraintes largement étudiées dans la littérature. Dans un premier temps, nous écrivons un modèle de satisfaction de contraintes et d'optimisation ainsi que les stratégies de recherche que nous avons envisagées. Ensuite, nous décrivons des contraintes supplémentaires visant à limiter l'espace de recherche des solutions. Enfin, nous présentons les résultats obtenus par ce modèle dans ces différentes configurations.

## 9.2 Modèle et stratégies de recherche

La programmation par contraintes a déjà été utilisée pour le problème de speed-dating avec contraintes additionnelles de limitation des déplacements dans (Lapègue et al., 2011). Dans cette référence, les variables utilisées représentent la femme rencontrée par un homme à une date donnée ou l'homme rencontré par une femme à une date donnée. Ce choix de variable permet de modéliser facilement les contraintes de déplacement des hommes.

Nous proposons un modèle de satisfaction de contraintes et d'optimisation pour le problème général de speed-dating basé sur les dates de rencontres. Ce choix de variables permet de modéliser facilement notre problème (en comparaison avec des variables représentant les participant(e)s rencontré(e)s à une date donnée) en utilisant des contraintes de type “AllDifferent” efficaces et très étudiées dans la littérature (voir (Beldiceanu et al., 2005)). Nous définissons pour cela les variables suivantes qui reprennent les notations standards de notre problème.

- $W_{\max}$  : le plus grand nombre de créneaux d'attentes des participants ;
- $t_{ij}$  : la date de début de la rencontre entre l'homme  $M_i \in \mathcal{M}$  et la femme  $F_j \in \mathcal{F}$  si  $p_{ij} = 1$  ( $t_{ij} = C_{ij} - 1$ ).

Le modèle s'écrit de la manière suivante.

$$\min W_{\max} \quad (9.1)$$

sous les contraintes :

$$\forall i \in \llbracket 0, m-1 \rrbracket, \text{ AllDifferent}\{t_{ij} \mid j \in \llbracket 0, n-1 \rrbracket, p_{ij} = 1\} \quad (9.2)$$

$$\forall j \in \llbracket 0, n-1 \rrbracket, \text{ AllDifferent}\{t_{ij} \mid i \in \llbracket 0, m-1 \rrbracket, p_{ij} = 1\} \quad (9.3)$$

$$\forall (i, j) \in \llbracket 0, m-1 \rrbracket \times \llbracket 0, n-1 \rrbracket \text{ tel que } p_{ij} = 1, \quad t_{ij} + 1 - d_{ij} \leq W_{\max} \quad (9.4)$$

$$\forall (i, j) \in \llbracket 0, n-1 \rrbracket \times \llbracket 0, n-1 \rrbracket \text{ tel que } p_{ij} = 1, \quad \text{Dom}(t_{ij}) = \llbracket r_{ij}, d_{ij} - 1 + UB \rrbracket \quad (9.5)$$

$$\text{Dom}(W_{\max}) = \llbracket LB, UB \rrbracket. \quad (9.6)$$

L'objectif (9.1) est de minimiser le nombre maximum d'attentes des participants. Les contraintes (9.2) et (9.3) stipulent que les dates de début de rencontres d'un participant doivent être toutes différentes.

La contrainte  $\text{AllDifferent}\{x_1, \dots, x_n\}$  indique que l'ensemble des valeurs affectées aux variables de  $\{x_i \mid i \in \llbracket 1, n \rrbracket\}$  sont différentes deux à deux (voir (Régin, 2011) par exemple). Les contraintes (9.4) permettent d'établir le lien entre l'objectif  $W_{\max}$  et les variables de décision. Les contraintes (9.5) et (9.6) définissent les domaines de définition des variables. En particulier, chacune des rencontres ne peut avoir lieu avant l'arrivée des deux participants. Les domaines des variables dépendent d'une borne inférieure  $LB$  et d'une borne supérieure  $UB$  pour l'objectif (nous utilisons les bornes définies dans les chapitres 6 et 7).

Pour ce modèle, nous proposons plusieurs stratégies de choix de variables et de valeur à affecter à la variable. Les stratégies de recherche que nous allons tester sont les suivantes.

**mdmv (MinDomainMinValue) :** on choisit la plus petite valeur de la variable dont le domaine des valeurs est le plus petit. Cette stratégie vise à trouver des solutions réalisables rapidement car les variables dont le domaine est le plus petit sont critiques vis-à-vis de la réalisabilité. Le choix de la valeur la plus petite assure que les solutions explorées en premier sont actives, on sait que les solutions actives dominent l'espace des solutions (cf. proposition 8.5.6) ce qui oriente la recherche vers des solutions prometteuses.

**mdso (MinDomainSmallestOccurence) :** on choisit la variable de plus petit domaine notée  $t_{ij}$ . Parmi les valeurs du domaine courant de cette variable, on choisit la valeur qui apparaît le moins dans les domaines des variables liées (*i.e.*, les variables correspondant aux rencontres des participants  $M_i$  et  $F_j$ ). Cette stratégie vise à la recherche de solutions réalisables sans recherche de qualité de celles-ci. Le risque de ne pas trouver de solution de bonne qualité est faible car la différence entre les meilleures bornes supérieures et inférieures est toujours très petite (4 au maximum cf. tableau 7.10).

**mwmd (MostWaitingMinValue) :** on choisit la variable correspondant à la rencontre dont les participants attendent le plus dans l'ordonnancement partiel des rencontres. Parmi les participants qui ont le plus grand nombre d'attente dans l'ordonnancement partiel, on choisit sa rencontre avec le(la) participant(e) qui a le plus de créneaux

d'attente. La valeur sélectionnée pour cette variable est la plus petite de son domaine. Cette stratégie privilégie la recherche de solutions de qualité, on considère de manière prioritaire les participants qui attendent le plus et on affecte à leur date de rencontre la plus petite date possible dans le but de limiter le nombre maximum de créneaux d'attente des participants.

### 9.3 Réduction du domaine de recherche

Dans cette section, nous proposons plusieurs contraintes additionnelles dérivées de règles de dominance (certaines ayant été présentées dans le chapitre 8) visant à limiter l'espace des solutions.

#### 9.3.1 Ordonnancement actif

On rappelle qu'un ordonnancement est actif si aucune rencontre ne peut être ordonnancée plus tôt sans retarder une autre rencontre ou violer des contraintes. Nous avons démontré que les ordonnancements actifs sont dominants pour notre problème (cf. proposition 8.5.6). Les contraintes logiques (9.7) garantissent la construction d'un ordonnancement actif.

$$\forall i \in \llbracket 0, m-1 \rrbracket, j \in \llbracket 0, n-1 \rrbracket \text{ t.q. } p_{ij} = 1, \forall k \in \llbracket r_{ij} + 1, s-1 \rrbracket, \forall k' \in \llbracket r_{ij}, k-1 \rrbracket, \\ t_{ij} = k \Rightarrow \text{card} \{ t_{\lambda\mu} \mid (\lambda, \mu) \in (\{i\} \times \llbracket 0, n-1 \rrbracket) \cup (\llbracket 0, m-1 \rrbracket \times \{j\}) \}, p_{\lambda\mu} = 1, t_{\lambda\mu} = k' \} \geq 1. \quad (9.7)$$

Les contraintes (9.7) signifient que si la rencontre entre l'homme  $M_i$  et la femme  $F_j$  (pour  $i \in \llbracket 0, m-1 \rrbracket$  et  $j \in \llbracket 0, n-1 \rrbracket$ ) a lieu à la date  $k$ , alors cette rencontre ne peut pas avoir lieu plus tôt, c'est-à-dire que pour toute date entre  $r_{ij}$  et  $k-1$  au moins un des participants  $M_i$  ou  $F_j$  a une rencontre à cette date.

La contrainte  $\text{card} \{ t_{\lambda\mu} \mid (\lambda, \mu) \in (\{i\} \times \llbracket 0, n-1 \rrbracket) \cup (\llbracket 0, m-1 \rrbracket \times \{j\}) \}, p_{\lambda\mu} = 1, t_{\lambda\mu} = k' \} \geq 1$  est classique en programmation par contraintes. La contrainte  $\text{count}(\text{value}, (\text{vars}_i)_{i \in \llbracket 1, n \rrbracket}, \text{relop}, \text{limit})$  (voir (Beldiceanu et al., 2005)) permet de modéliser les contraintes (9.7). Dans cette contrainte, *value* est une valeur numérique, *variables* est une liste de variables, *relop* est un opérateur de comparaison dans  $\{=, \neq, <, \leq, >, \geq\}$  et *limit* est une variable. Cette contrainte stipule que le nombre de variables vérifiant la condition  $\text{vars}_i \text{ relop limit}$  est égal à la valeur *value*.

#### 9.3.2 Symétries des participants

La gestion des symétries en PPC est un point essentiel car la suppression de celles-ci permet d'éviter de perdre du temps à explorer des parties symétriques de l'arbre de recherche. Le problème de casser les symétries en PPC a fait l'objet d'études spécifiques dans la littérature (par exemple (Gent et al., 2006; Walsh, 2006)). Une manière de casser les symétries se fait par l'ajout de contraintes qui éliminent des solutions symétriques (Puget, 1993). Dans la section 8.5.4, nous avons défini une règle de dominance cassant les symétries entre participants identiques (même date d'arrivée et même rencontres). Ces contraintes sont facilement adaptables au modèle de programmation par contraintes.

$$\forall i_1, i_2 \in \llbracket 0, m-1 \rrbracket \text{ t.q. } r_{i_1}^{(1)} = r_{i_2}^{(1)}, i_1 < i_2, \mathcal{F}_{M_{i_1}} = \mathcal{F}_{M_{i_2}}, j_0 = \min\{j \mid F_j \in \mathcal{F}_{M_{i_1}}\}, \quad t_{i_1 j_0} < t_{i_2 j_0} \quad (9.8)$$

$$\forall j_1, j_2 \in \llbracket 0, n-1 \rrbracket \text{ t.q. } r_{j_1}^{(2)} = r_{j_2}^{(2)}, j_1 < j_2, \mathcal{M}_{F_{j_1}} = \mathcal{M}_{F_{j_2}}, i_0 = \min\{i \mid M_i \in \mathcal{M}_{F_{j_1}}\}, \quad t_{i_0 j_1} < t_{i_0 j_2} \quad (9.9)$$

#### 9.3.3 Symétries des dates de rencontres

Nous définissons dans cette section une nouvelle règle de dominance permettant de casser des symétries entre dates de rencontres. On part du constat que certains créneaux de la soirée sont permutables sans modification de l'objectif ainsi que l'illustre l'exemple 9.3.1.

**Exemple 9.3.1.** Les deux solutions de la figure 9.1 sont symétriques par rapport aux dates de rencontres. En effet, on obtient la deuxième par permutation des rencontres ayant lieu aux dates 1, 2 et 3. On peut effectuer cette permutation sans changer le nombre d'attentes des participants à ces dates car tous les participants sont arrivés ( $\geq r_{\max} = 1$ ) et

aucun d'entre eux n'a fini ses rencontres ( $< d_{\min} = 4$ ). Éliminer les solutions redondantes par permutation de dates dans  $\llbracket 1, 3 \rrbracket$  dans cet exemple permet de diviser le nombre total de solutions par  $(d_{\min} - r_{\max})! = 3! = 6$ .

		$F_0$	$F_1$	$F_2$	$F_3$			$F_0$	$F_1$	$F_2$	$F_3$		
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1	1	$W_i^{(1)}$		0	0	1	1	$W_i^{(1)}$	
$M_0$	0	2	0	1	3	0	$M_0$	3	0	2	1	0	
$M_1$	1	3	1	2	4	0	$M_1$	1	2	3	4	0	
$M_2$	1	1	3		2	0	$M_2$	2	1		3	0	
$M_3$	1	4	2	3	1	0	$M_3$	4	3	1	2	0	
	$W_j^{(2)}$	1	0	0	0	$W_{\max} = 1$		1	0	0	0	$W_{\max} = 1$	

(a) Solution 1

(b) Solution 2

FIGURE 9.1 – Solutions symétriques par dates de rencontres

Nous avons vu sur un exemple que les dates de l'ensemble  $\llbracket r_{\max}, d_{\min} - 1 \rrbracket$  sont permutable sans modifier l'objectif. On peut généraliser ce résultat à l'intervalle  $\llbracket r_{\max}, d_{\min} - 1 + LB \rrbracket$  ainsi qu'énoncé dans la proposition 9.3.2.

**Règle de dominance 9.3.1.** On note  $\delta_{date}(\Omega)$  l'ensemble des solutions d'une instance de speed-dating pour lesquelles la première occurrence de chacune des dates de  $\llbracket r_{\max}, d_{\min} - 1 + LB \rrbracket$  (quand cet intervalle est non vide) apparaît par ordre croissant dans la suite  $((t_{ij})_{i \in \llbracket 0, m-1 \rrbracket})_{j \in \llbracket 0, n-1 \rrbracket}$ .

**Proposition 9.3.2.** L'ensemble  $\delta_{date}(\Omega)$  contient une solution optimale.

*Démonstration.* Soit une solution optimale d'une instance de speed-dating. On permute l'ensemble des dates de rencontres de manière à ce que les dates de  $\llbracket r_{\max}, d_{\min} - 1 + LB \rrbracket$  apparaissent par ordre croissant dans la suite  $((t_{ij})_{i \in \llbracket 0, m-1 \rrbracket})_{j \in \llbracket 0, n-1 \rrbracket}$ . Cette solution est valide : aucune rencontre n'a lieu avant l'arrivée des participants car tous les participants sont arrivés après la date  $r_{\max}$ . De plus, le nouvel objectif est inférieur ou égal à l'objectif de la solution de départ. En effet, les rencontres dont la date a été modifiée ont lieu au plus tard à la date  $d_{\min} - 1 + LB$ , le retard engendré par une rencontre  $(M_i, F_j)$  ayant lieu à cette date est égal à  $d_{\min} + LB - d_{ij} \leq LB$  (car  $d_{ij} \geq d_{\min}$ ). D'où le résultat.  $\square$

**Exemple 9.3.2.** La figure 9.2 présente une solution dominante pour l'instance de l'exemple 9.3.1. Dans cette solution,  $\llbracket r_{\max}, d_{\min} - 1 + LB \rrbracket = \llbracket 1, 4 \rrbracket$ , (on vérifie facilement que 1 est une borne inférieure de l'objectif car un seul homme est arrivé à la date 0 tandis que deux femmes sont à l'heure, (cf.  $LB_3$  à la section 6.5)). Cette solution est obtenue par permutation des dates 1 et 2 de la solution 9.1a.

		$F_0$	$F_1$	$F_2$	$F_3$	
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1	1	$W_i^{(1)}$
$M_0$	0	1	0	2	3	0
$M_1$	1	3	2	1	4	0
$M_2$	1	2	3		1	0
$M_3$	1	4	1	3	2	0
	$W_j^{(2)}$	1	0	0	0	$W_{\max} = 1$

FIGURE 9.2 – Solution dominante de  $\delta_{date}(\Omega)$

**Remarque 9.3.3.** Le groupe des permutations de l'ensemble des dates  $\llbracket r_{\max}, d_{\min} + LB - 1 \rrbracket$  est d'ordre  $(d_{\min} + LB - r_{\max})!$ . Ainsi, le cardinal des solutions dominantes  $\delta_{date}(\Omega)$  est  $(d_{\min} + LB - r_{\max})!$  fois inférieur au cardinal de l'ensemble total des solutions (pour une date de fin de soirée fixée). Sur l'ensemble des 300 instances que nous avons générées, la valeur  $(d_{\min} + LB - r_{\max})!$  est comprise entre  $2!$  et  $36! \approx 3,7 \times 10^{41}$ . Pour cela, il nous semble pertinent d'introduire la contrainte régulière (9.10).

L'ensemble  $\delta_{date}(\Omega)$  peut être décrit par la contrainte (9.10).

$$\text{regular}(((t_{ij})_{i \in \llbracket 0, m-1 \rrbracket})_{j \in \llbracket 0, n-1 \rrbracket}, \text{automaton } D). \quad (9.10)$$



La contrainte  $\text{regular}((vars_i)_{i \in \llbracket 1, n \rrbracket}, \text{automaton})$  (voir (Pesant, 2004)) assure que la séquence de variables  $(vars_i)_{i \in \llbracket 1, n \rrbracket}$  appartient au langage régulier défini par l'automate  $\text{automaton}$ . Dans notre cas, l'automate  $\text{automaton}D$  s'écrit de la manière suivante :

$$\begin{aligned} \text{automaton}D = & (\llbracket 0, r_{\max} - 1 \rrbracket \cup \llbracket d_{\min} + LB, s - 1 \rrbracket)^*(r_{\max})(\llbracket 0, r_{\max} \rrbracket \cup \llbracket d_{\min} + LB, s - 1 \rrbracket)^*(r_{\max} + 1) \dots \\ & \dots (\llbracket 0, d_{\min} + LB - 2 \rrbracket \cup \llbracket d_{\min} + LB, s - 1 \rrbracket)(d_{\min} + LB - 1)(\llbracket 0, s - 1 \rrbracket)^* \end{aligned}$$

L'automate  $\text{automaton}D$  décrit les suites possibles de variables qui peuvent être affectées à la suite de variables  $((t_{ij})_{i \in \llbracket 0, m-1 \rrbracket})_{j \in \llbracket 0, n-1 \rrbracket}$ . L'opérateur étoile se définit par  $A^* = \{x_1 x_2 \dots x_k \mid k \geq 0, \forall 0 \leq i \leq k, x_i \in A\}$ .

## 9.4 Résultats expérimentaux

Dans cette section, nous présentons les résultats obtenus par le modèle de programmation par contraintes décrit dans ce chapitre. Les bornes inférieures et supérieures que nous fournissons au modèle sont les meilleures que nous obtenons dans les chapitres 6 et 7, notées  $LB_{\text{best}}$  et  $UB_{\text{best}}$ . Nous testons donc les 166 instances qui ne sont pas résolues a priori par calcul de bornes inférieures et supérieures (cf. tableau 7.10). Nous avons utilisé le solveur de satisfaction de contraintes Choco3 (Choco-Team, 2010). Les tests ont été effectués sur un Intel Core i3-540 (3,06 GHz & 8 Go RAM) avec un temps limite de 30 minutes pour chaque instance.

### 9.4.1 Stratégies de recherche

Le tableau 9.1 présente le nombre d'instances résolues à l'optimum et le nombre d'instances pour lesquels la méthode a trouvé une solution optimale sans la prouver en 30 min selon la stratégie de recherche utilisée. De manière générale, on remarque que la programmation par contraintes a des difficultés à démontrer l'optimalité de solutions même pour des petites instances qui sont résolues facilement par la PLNE. Parmi les stratégies de recherche testées, la stratégie mdmv (MinDomainMinValue) s'avère la plus efficace sur l'ensemble des instances, en particulier les grandes instances.

	mdmv		mdso		mwmv		PLNE (Ag)		Total
	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	
14 p.	8	13	<b>9</b>	<b>14</b>	8	<b>14</b>	<b>14</b>	<b>14</b>	14
24 p.	18	23	<b>19</b>	<b>24</b>	<b>19</b>	<b>24</b>	<b>26</b>	<b>26</b>	26
40 p.	<b>21</b>	<b>32</b>	19	30	18	30	<b>34</b>	<b>34</b>	34
50 p.	<b>31</b>	<b>34</b>	30	33	26	30	<b>42</b>	<b>42</b>	42
70 p.	<b>28</b>	<b>40</b>	26	33	19	30	<b>48</b>	<b>48</b>	50
Total	<b>106</b>	<b>142</b>	103	134	90	128	<b>164</b>	<b>164</b>	166

Tableau 9.1 – Nombre de solutions optimales trouvées et démontrées

Le tableau 9.2 présente le nombre d'optima trouvés et le nombre d'optima démontrés en fonction de la proportion de femmes et d'hommes. La configuration mdso (MinDomainSmallestOccurrence) domine les autres configurations pour les instances carrées.

	mdmv		mdso		mwmv		PLNE (Ag)		Total
	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	
40%/60%	29	52	26	44	<b>34</b>	<b>57</b>	60	60	60
50%/50%	32	33	<b>34</b>	<b>35</b>	15	16	43	43	45
60%/40%	<b>45</b>	<b>57</b>	43	55	41	55	61	61	61

Tableau 9.2 – Nombre de solutions optimales trouvées et démontrées



Le tableau 9.3 présente les temps de calcul pour les instances résolues à l'optimum. Lorsque la méthode trouve (et prouve) l'optimalité d'une instance, le temps de calcul est très rapide quelle que soit la stratégie de recherche utilisée. Ce temps de calcul augmente légèrement avec la taille de l'instance.

	mdmv			mdso			mwmv		
	[0s,1s[	[1s,10s[	[10s,1800s[	[0s,1s[	[1s,10s[	[10s,1800s[	[0s,1s[	[1s,10s[	[10s,1800s[
14 p.	8			9			8		
24 p.	18			19			19		
40 p.	21			19			18		
50 p.	30	0	1	28	2		20	6	
70 p.	0	28		0	26		0	19	

Tableau 9.3 – Temps de résolution à l'optimum

### 9.4.2 Règles de dominance

Nous avons testé la stratégie mdmv (qui fournit globalement les meilleurs résultats) avec les contraintes additionnelles de la section 9.3. visant à réduire l'espace de recherche.

Dans le tableau 9.4, nous présentons le nombre d'instances résolues à l'optimum après ajout de contraintes de limitation aux ordonnancements actifs, de symétries par rapport aux participants et aux dates de rencontres. On ne constate pas d'amélioration significative du nombre d'optima trouvés. Avec la règle de dominance sur les symétries des participants, on trouve un optimum de plus pour les instances à 50 participants mais on en perd un pour les instances à 24 participants. En ce qui concerne les contraintes de symétries sur les dates, le solveur ne réussit pas à construire l'automate pour certaines instances à 50 participants et plus et pour lequel l'automate est trop grand.

	Actif		Symétries participants		Symétries dates		mdmv		Total
	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	
14 p.	8	13	8	13	8	13	8	13	14
24 p.	18	23	17	22	17	22	18	23	26
40 p.	21	32	21	<b>33</b>	21	<b>33</b>	21	32	34
50 p.	31	34	<b>32</b>	<b>35</b>	–	–	31	34	42
70 p.	28	40	28	39	–	–	28	40	50
Total	106	142	106	142	–	–	106	142	166

Tableau 9.4 – Solutions optimales trouvées et démontrées par taille

Dans le tableau 9.5, les nombres d'instances optimales trouvées et prouvées sont classés par proportion femmes/hommes. On constate qu'en ce qui concerne la configuration avec contraintes de symétries par rapport aux participants, un optimum supplémentaire est trouvé ou prouvé pour les instances non carrées tandis qu'un optimum est perdu pour les instances carrées.

	Actif		Symétries participants		mdmv		Total
	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	# Opt prouvés	# Opt trouvés	
40%/60%	29	52	<b>30</b>	52	29	52	60
50%/50%	32	33	31	32	32	33	45
60%/40%	45	57	45	<b>58</b>	45	57	61

Tableau 9.5 – Solutions optimales trouvées et démontrées par proportion femmes/hommes

Le tableau 9.6 présente les temps de calcul des optima obtenus pour les différentes règles de dominance testées. Quelle que soit la configuration, les temps de calculs sont dégradés par rapport à la configuration mdmv sans ajout de

contraintes. Il est possible que certaines solutions optimales non dominantes soient écartées au début de la recherche et que l'exploration soit conduite dans des mauvaises branches avant de finir par trouver une solution optimale dominante et le temps perdu n'est pas compensé par celui gagné à couper des branches ne contenant pas de solutions optimales.

	Actif			Symétries des participants			Symétries des dates			mdmv		
	[0,1[	[1,10[	[10,1800[	[0,1[	[1,10[	[10,1800[	[0,1[	[1,10[	[10,1800[	[0,1[	[1,10[	[10,1800[
14 p.	7	1		7	0	1	7	0	1	8		
24 p.	17	1		17			17			18		
40 p.	12	9		20	0	1	2	18	1	21		
50 p.	3	27	1	31	0	1				30	0	1
70 p.	0	27	1	0	28					0	28	

Tableau 9.6 – Temps de résolution à l'optimum

En conclusion, on peut dire que les règles de dominance étudiées n'ont pas permis d'améliorer le modèle de base. On aurait pu attendre une amélioration du nombre de preuves d'optimalité car l'ajout de règles de dominance réduit significativement l'arbre de recherche restant à explorer après l'obtention d'une solution optimale. Ce n'est cependant pas le cas ici. On peut supposer que les règles de dominance ne détectent pas toutes les symétries du problème.

## 9.5 Conclusion

Dans ce chapitre, nous avons proposé un modèle de satisfaction de contraintes et d'optimisation pour le problème de speed-dating ainsi que plusieurs stratégies de recherche pour ce modèle. Dans un but de limiter l'espace des solutions, nous avons ajouté des contraintes de dominance à ce modèle.

Les tests réalisés à ce jour montrent que cette modélisation n'est pas aussi efficace que nous pouvions l'espérer, les investigations doivent être encore poursuivies. Ce modèle est encore améliorable, d'autres stratégies de recherche et d'autres règles de dominance peuvent être implémentées.

Dans la partie suivante, nous développons une méthode approchée basée sur une recherche arborescente tronquée dans le but de mieux gérer l'exploration de l'arbre.



# Chapitre 10

## Recherche à divergence limitée (Limited Discrepancy Search)

### Sommaire

10.1 Introduction	107
10.2 Schéma de branchement et critère heuristique	108
10.3 Élagage de l'arbre	109
10.4 Expérimentations numériques	113
10.5 Conclusion	117

### 10.1 Introduction

La recherche à divergence limitée (Limited Discrepancy Search) est une méthode de résolution approchée décrite dans (Harvey et Ginsberg, 1995). L'idée de la méthode consiste à définir un arbre de recherche des solutions qui sera exploré suivant le degré d'adéquation avec un critère heuristique choisi. On décide d'explorer l'arbre de recherche de manière à s'éloigner de plus en plus des décisions de l'heuristique. La "divergence" (discrepancy) d'un nœud de l'arbre est une mesure de l'éloignement de ce nœud par rapport à la décision de base de l'heuristique. Pour une divergence fixée, on commence par l'exploration des branches qui débutent le plus haut dans l'arbre. Cette exploration permet de diversifier les solutions cherchées. La figure 10.1 présente l'ordre d'exploration d'un arbre de recherche à divergence limitée. La méthode de (Harvey et Ginsberg, 1995) est décrite pour la résolution de problèmes de décision mais se généralise facilement aux problèmes d'optimisation. Cette méthode a donné par exemple de bons résultats pour le problème de job-shop préemptif dans (Le Pape et Baptiste, 1999).

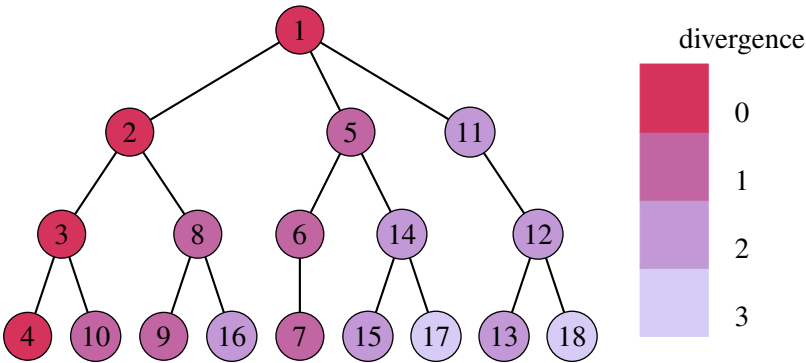


FIGURE 10.1 – Ordre d'exploration d'une recherche à divergence limitée

Dans ce chapitre, nous développons une recherche à divergence limitée pour le problème de speed-dating. Cette méthode nécessite une bonne heuristique pour guider la recherche rapidement vers une bonne solution. Nous disposons de plusieurs méthodes efficaces décrites dans le chapitre 7. Dans la section 10.2, nous présentons les stratégies de branchement et d'exploration utilisées pour la méthode. Dans la section 10.3, nous proposons plusieurs techniques permettant d'effectuer des coupes dans l'arbre de recherche. Nous y présentons une nouvelle règle d'élimination plus contraignante que la règle de dominance  $\delta_a(\Omega)$  (ordonnancements actifs). Nous explicitons le calcul de bornes inférieures destructives et de bornes supérieures déterminées au cours de la recherche. Enfin, nous définissons des classes d'équivalence de rencontres permettant de couper des branches équivalentes au fur et à mesure de la recherche. L'algorithme complet de cette méthode est présenté dans l'algorithme 7. Dans la section 10.4, nous présentons les résultats obtenus par cette méthode sur l'ensemble de nos 300 instances.

## 10.2 Schéma de branchement et critère heuristique

Le schéma de branchement que nous avons retenu est basé sur l'énumération des rencontres pouvant avoir lieu au premier créneau disponible. Les variables que nous utilisons représentent la date de début de chaque rencontre notée  $t_{ij}$  pour  $M_i \in \mathcal{M}$  et  $F_j \in \mathcal{F}$  tels que l'homme  $M_i$  et la femme  $F_j$  se rencontrent. Pour un ordonnancement partiel donné, on détermine la plus petite date à laquelle peut être ordonné un rendez-vous. Pour chaque rencontre dont les participants sont arrivés à cette date, on crée un nœud qui positionne cette rencontre à la date donnée. Pour un nœud donné  $N$ , si un de ses fils est un ordonnancement partiel contenant le rendez-vous entre  $M_i$  et  $F_j$  à la date  $t$ , alors pour l'ensemble des fils suivants de  $N$ , la date  $t$  est interdite pour ce rendez-vous.

Dans le chapitre 7 à la section 7.2, nous avons développé plusieurs heuristiques d'ordonnement en série permettant de trouver rapidement de premières solutions. Différents ordres de priorité classant les rencontres à affecter ont ainsi été définis. Au nœud racine de la résolution, on utilise l'ensemble des heuristiques du chapitre 7 pour fournir la meilleure borne supérieure valide à la méthode. Les temps de calculs de ces heuristiques sont négligeables devant la recherche arborescente. Le choix du critère heuristique dépend des calculs d'heuristiques à la racine. Le critère heuristique retenu pour la LDS est celui de l'heuristique de liste qui a fourni la meilleure borne supérieure. Par exemple, si la meilleure heuristique de liste est  $H_{wd}$ , l'ordre de priorité permettant de calculer la divergence d'un nœud se fera par l'ordre  $wd$ .

**Exemple 10.2.1.** La figure 10.2, présente un ordonnancement partiel représentant un nœud de l'arbre de recherche. Les rencontres grisées représentent les rencontres déjà ordonnées. On remarque qu'il est encore possible d'ordonner des rencontres à la date 1 (une croix représente une rencontre qui ne peut pas être ordonnée à la date 1). Les nombres entourés représentent l'ordre de priorité des rencontres à affecter. Dans la figure 10.2a, on suit l'ordre  $w$ , les rencontres de la femme  $F_1$  sont prioritaires car on sait qu'elle a un créneau d'attente à la date 0 quel que soit le reste de la solution car il n'est plus possible de placer des rendez-vous à la date 0. Dans la figure 10.2b, on emploie l'ordre lexicographique  $wd$ . Les rencontres de l'homme  $M_0$  et  $M_3$  sont prioritaires devant les autres car leur date d'échéance est la plus petite (3). Dans la figure 10.2c, on présente la priorité des rencontres pour l'ordre lexicographique  $wdm$ . Ici, l'ordre est le même que pour le critère précédent, ce n'est en général pas le cas pour des instances de taille plus grande.

	$F_0$	$F_1$	$F_2$	
$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	attentes
$M_0$	0	0 ① ②	0	0
$M_1$	0	1 × 0	0	
$M_2$	1	× ① ②	0	0
$M_3$	1	× ①	■	0
attentes		1	0	

(a) Ordre  $w$

	$F_0$	$F_1$	$F_2$	
$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	$e_j^{(2)}$
$M_0$	0	0 ① ③	0	3
$M_1$	0	1 × 0	0	
$M_2$	1	× ② ④	0	4
$M_3$	1	× ①	■	3
$e_i^{(1)}$		4	3	

(b) Ordre  $wd$

	$F_0$	$F_1$	$F_2$	
$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	nb. rencontres
$M_0$	0	0 ① ③	0	2
$M_1$	0	1 × 0	0	
$M_2$	1	× ② ④	0	3
$M_3$	1	× ①	■	2
nb. rencontres		4	2	

(c) Ordre  $wdm$

FIGURE 10.2 – Ordre de priorité des rencontres

La figure 10.2 présente donc trois possibilités de choix heuristiques parmi les 15 possibles (cf. chapitre 7). Les

ordres fournis par ces critères ne sont pas totaux. Le critère discriminant en cas de rencontres de même rang est celui de l'ordre lexicographique des femmes puis des hommes.

La figure 10.3 illustre le schéma de branchement que nous avons développé. L'ordre des fils courant correspond dans ce cas à l'ordre  $wdm$  (ou  $wd$ ) détaillé dans la figure 10.2c.

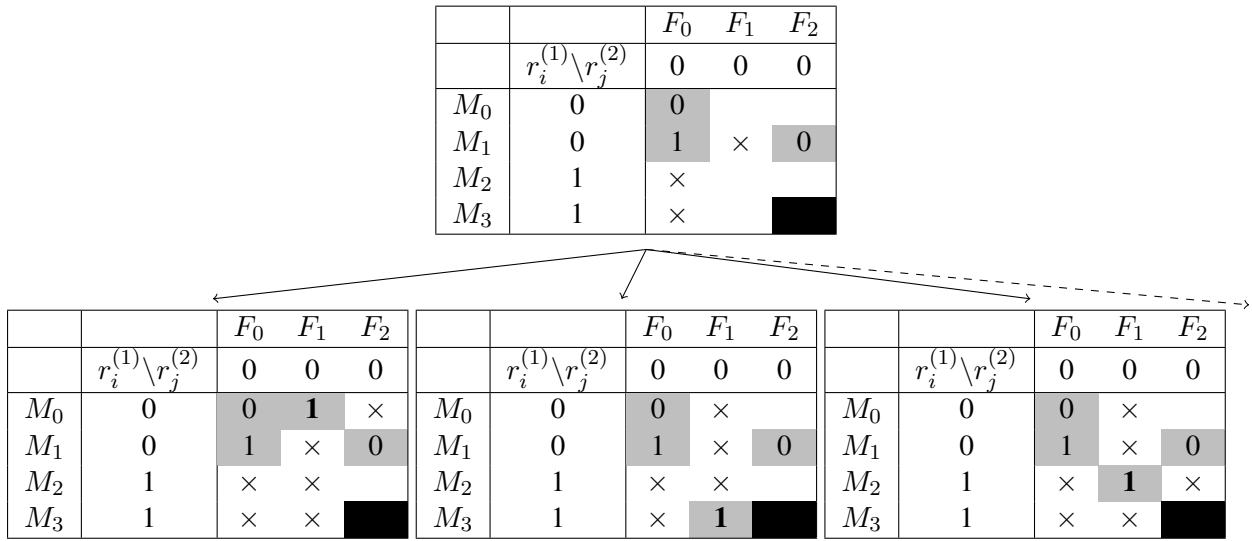


FIGURE 10.3 – stratégie de branchement

## 10.3 Élagage de l'arbre

Les méthodes de recherche arborescente nécessitent de bonnes coupes pour optimiser la recherche. Notamment, on utilise des calculs de bornes inférieures et supérieures et on exploite des règles de dominance pour couper le plus de branches possible. Dans le cas du problème de speed-dating, la recherche de coupes est d'autant plus importante que le nombre de solutions optimales du problème peut être très grand. Une instance à quatre femmes et quatre hommes peut avoir plus d'un million de solutions optimales (pour un objectif optimal égal à 2). Parmi ces solutions, beaucoup sont similaires, on cherche donc à déterminer des critères d'élimination afin d'élaguer des branches dont les solutions sont équivalentes (cf. définition 10.3.4) aux solutions d'un autre nœud déjà exploré.

### 10.3.1 Ordonnancement maximalement-actif

Afin de limiter l'exploration de l'arbre de recherche, on se sert de règles de dominance présentées dans le chapitre 8 section 8.5. On coupe les nœuds dont l'ordonnancement n'est pas actif. Cette règle de dominance va dans le sens de l'exploration de l'arbre puisqu'on positionne en priorité des rencontres à la date la plus petite.

Afin de couper plus de branches, on pose également une règle d'élimination plus forte. On définit pour cela les ordonnancements "maximalement-actifs".

**Définition 10.3.1.** On dit qu'un ordonnancement  $s$  est *maximalement actif* si pour chaque date  $t \in \llbracket 0, C_{\max}(s) \rrbracket$ , il n'existe pas d'ordonnancement  $s'$  tels que les rencontres de la date 0 à la date  $t - 1$  de  $s$  et  $s'$  soient les mêmes, et le nombre de rendez-vous positionnés à la date  $t$  dans l'ordonnancement  $s'$  soit strictement supérieur au nombre de rendez-vous positionnés à la date  $t$  dans  $s$ . On note l'ensemble des ordonnancements maximalement actifs :  $\delta_{ma}(\Omega)$ .

**Exemple 10.3.1.** Dans la figure 10.4, on propose deux ordonnancements distincts pour une même instance à trois femmes et trois hommes. La figure 10.4a représente un ordonnancement actif pour cette instance. En effet, aucun des rendez-vous ne peut être placé plus tôt sans en déplacer d'autres. Cet ordonnancement n'est pas maximalement actif, car le nombre de rencontres positionnées à la date 1 n'est pas maximal. Une fois les rencontres à la date 0 placées, on peut positionner trois rencontres à la date 1 comme dans la figure 10.4b. Dans cette dernière solution, une fois les

rencontres positionnées aux dates 0 et 1, on peut positionner au maximum deux rencontres à la date 2, ce qui est le cas ici et enfin, il reste une rencontre qu'on place à la date 3.

		$F_0$	$F_1$	$F_2$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1
$M_0$	0	0		2
$M_1$	0	1	0	3
$M_2$	0	3	2	1

(a) ordonnancement actif non maximalement actif

		$F_0$	$F_1$	$F_2$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	1
$M_0$	0	0		1
$M_1$	0	1	0	2
$M_2$	0	2	1	3

(b) ordonnancement maximalement actif

FIGURE 10.4 – Exemples d'ordonnancements actifs et maximalement actifs

**Proposition 10.3.2.** *Un ordonnancement maximalement actif est actif. C'est-à-dire que  $\delta_{ma}(\Omega) \subset \delta_a(\Omega)$ .*

*Démonstration.* On considère un ordonnancement maximalement actif. Dans cet ordonnancement, aucune rencontre ne peut être positionnée plus tôt. Supposons le contraire, si une rencontre positionnée à la date  $k$  peut être positionnée plus tôt à la date  $k'$  avec  $k' < k$ , cela signifie que le nombre de rencontres positionnées à la date  $k'$  n'est pas maximal, ce qui contredit la définition d'ordonnancement maximalement actif.  $\square$

**Conjecture 10.3.2.** L'ensemble des ordonnancements maximalement actifs optimaux pour le critère  $W_{\max}$  est non vide.

Nous travaillons sur la démonstration de cette conjecture. Nous n'avons trouvé aucun contre-exemple à ce jour infirmant cette propriété, que ce soit parmi les 300 instances que nous avons générées ou d'autres instances fabriquées à la main. Nous l'utilisons donc comme règle d'élimination dans l'algorithme de la LDS.

### 10.3.2 Bornes inférieures et supérieures

L'algorithme détermine une borne inférieure globale notée  $LB$  à la racine qui est le maximum entre  $LB_2$  et  $LB_3$  (cf. chapitre 6). Cette borne inférieure est un critère d'arrêt de l'algorithme lorsque plus aucune branche n'admet une solution de meilleur objectif que celle-ci.

Également au nœud racine, l'algorithme détermine une borne supérieure globale notée  $UB$ . L'ensemble des heuristiques du chapitre 7 sont lancées et  $UB$  est l'objectif de la meilleure solution trouvée. Les domaines de variables sont ainsi déduits de  $UB$ . On cherche une solution d'objectif strictement inférieur à  $UB$ , ainsi le domaine d'une variable  $t_{ij}$  (date de rencontre entre l'homme  $M_i$  et la femme  $F_j$ ) qui est  $\llbracket r_{ij}, d_{ij} + UB - 2 \rrbracket$ .

En chaque nœud développé, une borne inférieure destructive est calculée. Il s'agit de tester si  $UB - 1$  est une borne inférieure. Pour cela, on exploite les contraintes de disjonction des rencontres de chaque participant. Chaque participant ne peut effectuer au maximum qu'une rencontre par créneau, chacune de ses dates de rencontres doit donc être différente. Il est possible de déterminer, en temps polynomial, l'ensemble des dates qui ne peuvent pas être affectées à chaque rencontre d'un participant sans violer cette contrainte de non-ubiquité. Il suffit d'effectuer une recherche de l'ensemble des arêtes dans un graphe biparti appartenant à un couplage maximal. Ce problème se résout en temps polynomial (Régin, 1994) et (Tassa, 2012, 2013). Si à la suite de ce filtrage de variables, le domaine d'une variable est vide alors il n'existe pas de solution valide pour l'objectif  $UB - 1$  et le nœud courant est coupé, sinon il ne se passe rien. Cette technique de filtrage du domaine des variables est largement utilisée en programmation par contraintes dans le cadre de la contrainte de type "AllDifferent". À noter que cette technique procure plus que la détection d'in-faisabilité puisqu'elle réduit les domaines des variables.

Dans la recherche arborescente, la recherche d'une meilleure borne supérieure est régulièrement effectuée. Il s'agit de déterminer si  $UB - 1$  est une borne supérieure. On met en place une heuristique de couplage : pour chaque date, on résout un problème d'affectation entre les hommes et les femmes disponibles à cette date, le poids associé à chaque rencontre est le nombre de dates encore disponibles pour cette rencontre. Le calcul de cette borne supérieure est uniquement basé sur la recherche d'une solution réalisable. Si une solution est trouvée, la borne supérieure globale

est mise à jour  $UB \leftarrow UB - 1$  ainsi que tous les domaines des variables. La fréquence de calcul de cette solution dépend du nombre de rencontres de l'instance. Ce calcul se fait toutes les  $\lfloor |\mathcal{Meet}|/\alpha \rfloor$  où  $\alpha$  est un paramètre à définir et  $\mathcal{Meet}$  est l'ensemble des rencontres à planifier.

### 10.3.3 Gestion des symétries

La gestion des symétries est cruciale pour notre problème car le nombre de solutions qu'on peut qualifier de "similaires" est très important. En effet, beaucoup de participants ont les mêmes caractéristiques et peuvent échanger leur dates de rendez-vous deux à deux. Nous avons défini des classes d'équivalence entre les ordonnancements afin de détecter facilement lors de la recherche des branches qui vont produire des solutions redondantes et ainsi de couper le plus de branches possibles. Dans un premier temps, on définit les participants équivalents.

**Définition 10.3.3.** Pour une instance donnée, on dit que deux participants sont *équivalents* si on peut les échanger et obtenir la même instance par permutations d'autres participants.

*Remarque 10.3.3.* En particulier, deux participants équivalents ont la même date d'arrivée et le même nombre de rencontres interdites.

#### Exemple 10.3.4.

Dans l'instance représentée par la figure 10.5, les classes d'équivalence des hommes sont :  $\{M_0, M_1\}$  ;  $\{M_2, M_3\}$  et  $\{M_4\}$ . Les classes d'équivalence des femmes sont :  $\{F_0\}$  ;  $\{F_1, F_2\}$  et  $\{F_3, F_4\}$ . En effet, les hommes  $M_0$  et  $M_1$  sont équivalents car si on les permute, on peut retrouver la même instance en permutant les hommes  $M_2$  et  $M_3$  et en permutant les femmes  $F_1$  et  $F_2$ . Pour les mêmes raisons, les femmes  $F_1$  et  $F_2$  sont équivalentes de même que les hommes  $F_2$  et  $F_3$ .

		$F_0$	$F_1$	$F_2$	$F_3$	$F_4$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	1	1
$M_0$	0	■	■	■		
$M_1$	0	■	■	■		
$M_2$	0		■	■		
$M_3$	0		■	■		
$M_4$	1				■	■

FIGURE 10.5 – Illustration de participants équivalents

La définition 10.3.3 de participants équivalents ne permet pas de déterminer les classes d'équivalence de participants facilement. Il n'est pas nécessaire de tester toutes les permutations possibles de participants pour déterminer ceux qui sont équivalents, cela serait trop coûteux en terme de temps de calcul et inutile. Nous proposons dans l'algorithme 6 une procédure permettant de construire ces classes de manière plus rapide. Cet algorithme 6 s'exécute en  $O((m+n)(m^2+n^2))$ . La boucle **Répéter** de la ligne 3 est exécutée au maximum  $m+n$  fois (nombre total de participants). Chacune des boucles **Pour** est répétée au maximum  $m^2$  fois pour celle de la ligne 4 (nombre maximal de comparaisons entre les hommes) et  $n^2$  fois pour celle de la ligne 8.



**Algorithme 6** Construction des classes d'équivalence des participants

**Paramètres :**  $\mathcal{M}, \mathcal{F}, (r_i^{(1)})_{i \in \llbracket 0, m-1 \rrbracket}, (r_j^{(2)})_{j \in \llbracket 0, n-1 \rrbracket}$

- 1:  $\mathcal{M} = \bigcup_{l=1}^t C_M^l$  partition de  $\mathcal{M}$  en sous-ensembles d'hommes arrivant à la même date.
- 2:  $\mathcal{F} = \bigcup_{k=1}^s C_F^k$  partition de  $\mathcal{F}$  en sous-ensembles de femmes arrivant à la même date.
- 3: **Répéter**
- 4:   **Pour chaque**  $C_M^l \subset \mathcal{M}$  **faire**
- 5:     Partitionner  $C_M^l$  en sous-ensembles tels que deux hommes de  $C_M^l$  sont dans le même sous-ensemble si et seulement si ils ont le même nombre de rencontres dans chacun des sous-ensembles  $C_F^k$  courants
- 6:   **Fin Pour**
- 7:   Mettre à jour la partition de  $\mathcal{M}$  avec ces sous-ensembles
- 8:   **Pour chaque**  $C_F^k \subset \mathcal{F}$  **faire**
- 9:     Partitionner  $C_F^k$  en sous-ensembles tels que deux femmes de  $C_F^k$  sont dans le même sous-ensemble si et seulement si elles ont le même nombre de rencontres dans chacun des sous-ensembles  $C_M^l$  courants
- 10:   **Fin Pour**
- 11:   Mettre à jour la partition de  $\mathcal{F}$  avec ces sous-ensembles
- 12: **jusqu'à** ce que les partitions de l'ensemble des hommes et de l'ensemble des femmes restent inchangées.
- 13: Les partitions de  $\mathcal{F}$  et  $\mathcal{M}$  définissent les classes d'équivalence des participants

**Exemple 10.3.5.** Déroulons l'algorithme 6 sur l'exemple précédent (cf. figure 10.5). Les partitions de  $\mathcal{M}$  et  $\mathcal{F}$  initiales scindent les participants par date d'arrivée. Ainsi,  $C_M^1 = \{M_0, M_1, M_2, M_3\}$  et  $C_M^2 = \{M_4\}$ . De même,  $C_F^1 = \{F_0, F_1, F_2\}$  et  $C_F^2 = \{F_3, F_4\}$ .

Lors de la première itération, on considère  $C_M^1$ . On partitionne  $C_M^1$  en deux sous-ensembles  $\{M_0, M_1\}$  et  $\{M_2, M_3\}$  car  $M_0$  et  $M_1$  effectuent exactement une rencontre avec une femme de  $C_F^1$  et exactement deux rencontres avec une femme de  $C_F^2$ , alors que  $M_2$  et  $M_3$  ont exactement deux rencontres prévues avec une femme de  $C_F^1$  et deux rencontres avec les femmes de  $C_F^2$ . Il n'est pas nécessaire de chercher à partitionner  $C_M^2$  qui ne contient qu'un homme. Lors de la mise à jour de la partition de  $\mathcal{M}$ , on obtient trois sous-ensembles qu'on renomme  $C_M^1 = \{M_0, M_1\}$ ,  $C_M^2 = \{M_2, M_3\}$  et  $C_M^3 = \{M_4\}$ .

On s'occupe maintenant des sous-ensembles de femmes. On partitionne  $C_F^1$  en  $\{F_0\}$  et  $\{F_1, F_2\}$ . En effet,  $F_0$  n'effectue aucune rencontre avec les hommes de  $C_M^1$ , deux rencontres avec les hommes de  $C_M^2$  et une rencontre avec un homme de  $C_M^3$ , tandis que  $F_1$  et  $F_2$  effectuent exactement une rencontre dans chacun des ensembles  $C_M^1$ ,  $C_M^2$  et  $C_M^3$ . L'ensemble  $C_F^2 = \{F_3, F_4\}$  ne change pas car ses deux femmes effectuent exactement les mêmes rencontres. On met alors la partition de  $\mathcal{F}$  à jour, on a  $C_F^1 = \{F_0\}$ ,  $C_F^2 = \{F_1, F_2\}$  et  $C_F^3 = \{F_3, F_4\}$ .

Dans la deuxième itération, les partitions de  $\mathcal{M}$  et  $\mathcal{F}$  restent inchangées et l'algorithme s'arrête.

**Définition 10.3.4.** Pour une instance donnée, on dit que deux *ordonnancements* sont équivalents si on peut obtenir l'un à partir de l'autre par permutations de participants équivalents.

**Exemple 10.3.6.** On présente dans la figure 10.6 deux ordonnancements équivalents pour la même instance.

		$F_0$	$F_1$	$F_2$	$F_3$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	1
$M_0$	0	0	1	2	3
$M_1$	0	2	0	3	4
$M_2$	0	1	3	0	2
$M_3$	1		2	4	1

(a) première solution

		$F_0$	$F_1$	$F_2$	$F_3$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0	1
$M_0$	0	1	0	3	2
$M_1$	0	0	2	1	3
$M_2$	0	2	3	0	4
$M_3$	1		4	2	1

(b) deuxième solution

FIGURE 10.6 – Exemple d'ordonnancements équivalents

Afin de montrer que les ordonnancements 10.6a et 10.6b sont équivalents, on partitionne d'abord l'ensemble des femmes et l'ensemble des hommes en participants équivalents. Les hommes  $M_0$ ,  $M_1$  et  $M_2$  sont équivalents car ils arrivent à la même date et effectuent les mêmes rencontres. L'homme  $M_3$  est seul dans sa classe d'équivalence. La

femme  $F_0$  est également seule dans sa classe d'équivalence car elle n'a qu'une rencontre interdite.  $F_1$  et  $F_2$  sont équivalentes car elles rencontrent tous les hommes et arrivent à la même date. La femme  $F_3$  est seule dans sa classe d'équivalence car elle est seule à arriver à la date 1.

On obtient la deuxième solution par permutations  $(M_0, M_1, M_2)$  et  $(F_1, F_2)$ .

**Remarque 10.3.7.** Deux ordonnancements équivalents ont la même valeur de fonction objectif car on ne fait que permuter des plannings de participants équivalents qui ont la même date d'échéance. C'est pour cela que ne conserver qu'un représentant par classe d'équivalence ne retire pas de solutions optimales dans la recherche arborescente.

L'objectif est de couper lors de la recherche arborescente les nœuds qui vont produire des ordonnancements équivalents afin de diminuer la taille de l'arbre. Pour cela, on définit la notion de zones d'équivalence pour un ordonnancement partiel.

**Définition 10.3.5.** Soit un ordonnancement partiel pour une instance donnée, on dit que deux participants sont *équivalents relativement à l'ordonnancement partiel*, si on peut les échanger et obtenir le même ordonnancement partiel par permutations d'autres participants.

**Définition 10.3.6.** Soit un ordonnancement partiel pour une instance donnée, on dit que deux *rencontres* non instanciées sont *équivalentes relativement à l'ordonnancement partiel*, si les hommes (resp. femmes) de ces rencontres sont équivalents (resp. équivalentes) relativement à l'ordonnancement partiel.

**Exemple 10.3.8.** La figure 10.7a présente les zones d'équivalence d'une instance à la racine, elles sont simplement délimitées par les participants équivalents. Dans la figure 10.7b, les zones ont été mises à jour en fonction de l'ordonnancement partiel des dates zéros. Les hommes  $M_1$  et  $M_2$  restent équivalents car si on les permute et qu'on permute les femmes  $F_1$  et  $F_2$ , on obtient le même ordonnancement. Dans la figure 10.7c, les zones ont été mises à jour en fonction de l'ordonnancement partiel des rencontres aux dates 0 et 1. Ici, les zones ne contiennent plus qu'une rencontre car aucun participant n'est équivalent.

		$F_0$	$F_1$	$F_2$			$F_0$	$F_1$	$F_2$			$F_0$	$F_1$	$F_2$
	$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0		$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0		$r_i^{(1)} \setminus r_j^{(2)}$	0	0	0
$M_0$	0				$M_0$	0		Z1		$M_0$	0		1	Z1
$M_1$	0	Z1		Z2	$M_1$	0		0	Z3	$M_1$	0		1	Z2
$M_2$	0				$M_2$	0	Z2	Z3	0	$M_2$	0	Z3	Z4	0
$M_3$	1			Z3	$M_3$	1			Z4	$M_3$	1			Z5

(a) zones à la racine

(b) zones à la date 0

(c) zones à la date 1

FIGURE 10.7 – Zones d'équivalence

La définition de zones en chaque nœud dépendant de l'ordonnancement partiel permet d'éviter la création de nœuds redondants. En effet, il n'est pas utile de positionner deux rencontres de la même zone à la même date dans deux nœuds fils différents car cela produirait des ordonnancements équivalents.

## 10.4 Expérimentations numériques

Dans cette section, nous présentons les résultats obtenus par la LDS sur l'ensemble des 166 instances pour lesquelles la meilleure borne inférieure est différente de la meilleure borne supérieure et qui ne sont donc pas résolues à la racine.

### 10.4.1 Performance des coupes

Dans un premier temps, nous nous intéressons à montrer l'apport des coupes que nous avons établies. Le tableau 10.1 présente le nombre d'optima trouvés par la LDS lorsqu'aucune coupe n'est apportée, puis lors de l'ajout d'une coupe

**Algorithme 7** Recherche à divergence limitée

---

```

1:  $LB \leftarrow LB_{best}$ 
2:  $UB \leftarrow UB_{best}$ 
3: Définition des variables :  $\forall i \in \mathcal{M}, j \in \mathcal{F}$ , t.q.  $p_{ij} = 1, t_{ij} \in \llbracket r_{ij} + 1, d_{ij} + UB - 2 \rrbracket$ 
4:  $F \leftarrow$  nouvelle file de priorité
5: Insérer le nœud racine dans  $F$ 
6: Tant que  $LB < UB$  et  $F \neq \emptyset$  faire
7:   Défiler le sommet  $n$  de  $F$ 
8:   //Mise à jour des domaines des variables de  $n$ 
9:    $\forall i \in \mathcal{F}, j \in \mathcal{M}$ , t.q.  $p_{ij} = 1, \text{Dom}(t_{ij}) \leftarrow \text{Dom}(t_{ij}) \cap \llbracket r_{ij}, d_{ij} + UB - 2 \rrbracket$ 
10:  Si le domaine d'une variable est vide Alors
11:    retour en 6
12:  Fin Si
13:  Filtrage de tous les domaines des variables par la contrainte "AllDifferent"
14:  Si le nœud  $n$  n'admet pas de solution Alors
15:    retour en 6
16:  Fin Si
17:  //Traitement d'un nœud feuille
18:  Si toutes les variables sont instantiées Alors
19:    Calcul de l'objectif de la solution associée :  $UB_{node}$ 
20:     $UB \leftarrow UB_{node}$ 
21:    Retour en 6
22:  Fin Si
23:  //Calcul d'une borne supérieure
24:  Si  $\lfloor |\mathcal{Meet}|/\alpha \rfloor$  divise iteration Alors
25:    Calcul d'une borne supérieure  $UB_{node}$  par heuristique de couplage.
26:    Si  $UB_{node} < UB$  Alors
27:       $UB \leftarrow UB_{node}$ 
28:      //Mise à jour des domaines des variables de  $n$ 
29:       $\forall i \in \mathcal{M}, j \in \mathcal{F}$ , t.q.  $p_{ij} = 1, \text{Dom}(t_{ij}) \leftarrow \text{Dom}(t_{ij}) \cap \llbracket r_{ij}, d_{ij} + UB - 2 \rrbracket$ 
30:    Fin Si
31:  Fin Si
32:  //On coupe le nœud si l'ordonnancement n'est pas super-actif
33:   $d \leftarrow$  plus petite date à laquelle peut être affectée une rencontre
34:  Si  $d$  est plus grand que la précédente date instantiée dans  $n$  Alors
35:    Si l'ordonnancement partiel de  $n$  n'est pas super actif à la date  $d - 1$  Alors
36:      retour en 6
37:    Fin Si
38:  Fin Si
39:  //Création des nœuds fils
40:  Définition de l'ensemble des zones  $Z$ 
41:  Pour chaque zone  $z$  de  $Z$  faire
42:    Si planifier une rencontre de  $z$  ne provoque pas un ordonnancement non actif Alors
43:      créer un nœud fils où cette rencontre est planifiée à la date  $d$ 
44:    Fin Si
45:    Trier les nœuds créés par ordre de priorité des rencontres nouvellement planifiées
46:    Insérer les nœuds créés dans  $F$  par ordre de priorité
47:  Fin Pour
48: Fin Tant que

```

---

à la fois. Dans ce tableau, on présente également les temps moyens d'obtention des solutions optimales lorsque celles-ci ont été trouvées<sup>1</sup>. Certains temps de calcul moyens sont nuls car les solutions optimales trouvées dans ce cas ont été trouvées par le calcul d'une heuristique à la racine. Par défaut, la divergence maximale a été fixée à 2 car des tests préliminaires ont montré qu'une solution optimale se trouvait toujours dans un nœud de divergence au plus égale à 2. Ce tableau montre que l'ajout de bornes inférieures, bornes supérieures et de zones de symétrie améliorent la convergence vers des solutions optimales. La règle d'élimination d'ordonnancement non maximalement-actifs dégrade cependant la recherche. Il se peut que la vérification d'un ordonnancement maximalement actif ne se fasse pas suffisamment tôt dans l'arbre puisqu'on attend d'avoir affecté toutes les rencontres possibles pour une date avant de vérifier si ce nombre de rencontres est maximal. Il est envisageable d'améliorer ce point en déterminant plus tôt pour une date donnée toutes les rencontres auxquelles elle ne peut être affectée. Il suffit pour cela de déterminer l'ensemble des arêtes pouvant appartenir à un couplage maximum dans un graphe biparti (voir (Régis, 1994) ou (Tassa, 2012, 2013)). Certaines instances ne sont pas résolues à l'optimum, cela est dû à l'explosion en mémoire de la méthode avant la fin de la recherche de l'arbre.

	$\emptyset$		Bornes inf.		Max. actif		Bornes sup.		Zones de symétrie	
	optima trouvés	temps de calcul	optima trouvés	temps de calcul	optima trouvés	temps de calcul	optima trouvés	temps de calcul	optima trouvés	temps de calcul
14p.	14/14	0,13s	14/14	0,02s	13/14	0s	14/14	0,52s	14/14	0,31s
24p.	23/26	22,05s	<b>25/26</b>	0,51s	19/26	1,68s	23/26	18,45s	24/26	15,01s
40p.	16/34	347,36s	<b>31/34</b>	79,65s	13/34	12,59s	23/34	160,11s	19/34	237,86s
50p.	13/42	273,87s	21/42	17,04s	10/42	20,44s	<b>31/42</b>	58,86s	13/42	90,94s
70p.	2/50	0s	11/50	61,41s	2/50	0s	<b>13/50</b>	53,07s	2/50	156,35s
Total	68/166		102/166		57/166		104/166		72/166	

Tableau 10.1 – Proportion d'optima trouvés et temps moyens d'obtention des optima

### 10.4.2 Version avec coupes

Dans le tableau 10.2, nous présentons le nombre de solutions optimales trouvées par la méthode lorsque toutes les coupes qui ont apporté une amélioration à la version de base sont ajoutées, c'est à dire qu'on ajoute le calcul de bornes inférieures et supérieures au cours de la recherche ainsi que le découpage en zones de symétrie. Le paramètre  $\alpha$  affectant la fréquence a été fixé à 20, c'est-à-dire qu'une recherche de borne supérieure est effectuée toutes les  $\lfloor |\mathcal{Meet}|/20 \rfloor$  itérations. La conjonction de toutes ces coupes permet de trouver l'ensemble des optima de toutes les instances jusqu'à 50 participants et permet de trouver près de 75% des solutions optimales pour les instances à 70 participants. La deuxième colonne du tableau contient les temps moyens d'obtention d'une solution optimale (de même que dans le tableau 10.1).

	optima trouvés	temps de calcul
14p.	14/14	0,01s
24p.	26/26	0,41s
40p.	34/34	1,59s
50p.	42/42	6,28s
70p.	37/50	57,05s
Total	153/166	

Tableau 10.2 – Proportion d'optima trouvés et temps moyens d'obtention des optima dans la recherche avec bornes inférieures, supérieures et zones de symétrie

Dans le tableau 10.3, nous présentons le nombre d'instances pour lesquelles la méthode termine l'exploration de l'arbre tronqué, et les temps moyens de calcul. À partir de 40 participants, l'arbre à explorer est trop grand et la

<sup>1</sup>Ces temps ne correspondent pas aux temps d'exécution de la méthode complète mais seulement des temps d'obtention d'une solution optimale.

méthode explose en mémoire avant la fin des calculs pour certaines instances.

Taille	Instances résolues	Temps moyens
14p.	14/14	0,27s
24p.	26/26	8,38s
40p.	27/34	337,32s
50p.	37/42	6,93s
70p.	28/50	65,19s

Tableau 10.3 – Exploration complète de l’arbre

Nous comparons dans le tableau 10.4 les temps de calcul de la LDS par rapport à ceux de la PLNE (dans sa meilleure configuration : avec bornes inférieure et supérieure, solution de départ et contraintes agrégées cf. tableau 8.6). Nous ne comparons que les instances pour lesquelles la LDS a trouvé une solution optimale et dont la meilleure borne inférieure est égale à l’optimum. Ainsi, on ne considère que des instances pour lesquelles la PLNE n’a pas à prouver l’optimalité. Nous ne disposons pas des temps d’obtention d’une solution optimale de la PLNE, c’est pour cela que nous n’ajoutons pas ces instances, il serait fortement intéressant d’avoir une estimation de ce temps. Il est intéressant de noter les bonnes performances de la LDS pour les instances carrées en particulier à partir de 50 et 70 participants qui sont par ailleurs les instances les plus difficiles à résoudre pour la PLNE. Il est surprenant que les résultats de la LDS ne soient pas compétitifs avec ceux de la PLNE pour les instances non carrées car elles sont a priori plus faciles. En effet leur objectif étant assez grand, le nombre de solutions optimales est plus important pour ce type d’instances.

Taille et proportion	LDS	PLNE	# instances
14 p.			
40%/60%	0,07s	<b>0,04s</b>	1
60%/40%	0,05s	<b>0,02s</b>	1
24 p.			
40%/60%	1,32s	<b>0,17s</b>	3
50%/50%	<b>0,18s</b>	0,22s	4
60%/40%	0,94s	<b>0,13s</b>	5
40 p.			
40%/60%	2,81s	<b>1,58s</b>	4
50%/50%	<b>0,52s</b>	3,21s	5
60%/40%	2,60s	<b>1,56s</b>	7
50 p.			
40%/60%	10,11s	<b>3,82s</b>	7
50%/50%	<b>4,99s</b>	20,36s	13
60%/40%	7,68s	<b>3,28s</b>	9
70 p.			
40%/60%	74,94s	<b>28,32s</b>	7
50%/50%	<b>50,20s</b>	853,22s	10
60%/40%	64,07s	<b>22,94s</b>	5

Tableau 10.4 – Comparaison LDS/PLNE – temps moyens de résolution

### 10.4.3 Fréquence de calcul d’une borne supérieure

Dans cette section, on s’intéresse à l’impact de la fréquence de calcul d’une borne supérieure dans la recherche. Nous comparons les configurations dans lesquelles la fréquence de calcul d’une borne supérieure est  $\lfloor |\mathcal{Meet}|/\alpha \rfloor$  avec  $\alpha \in \{10, 20, 40\}$ . Le tableau 10.5 présente le nombre d’optima trouvés selon les différentes configurations en moins de 30 min. On remarque que la fréquence de calcul d’une borne supérieure a un impact non négligeable sur l’obtention

de solutions optimales. Pour une fréquence avec  $\alpha = 40$ , on gagne 4 solutions optimales par rapport à une fréquence avec  $\alpha = 20$ . Par ailleurs, les temps moyens pour trouver une solution optimale sont similaires, l'augmentation de recherche de borne supérieure ne pénalise pas trop la méthode.

	fréquence $\alpha = 40$		fréquence $\alpha = 20$		fréquence $\alpha = 10$	
	optima trouvés	temps de calcul	optima trouvés	temps de calcul	optima trouvés	temps de calcul
14p.	14/14	0,01s	14/14	0,01s	14/14	0,01s
24p.	26/26	0,43s	26/26	0,41s	26/26	0,42s
40p.	34/34	1,66s	34/34	1,59s	34/34	1,77s
50p.	42/42	4,85s	42/42	6,28s	42/42	7,73s
70p.	38/50	49,86s	37/50	57,05s	33/50	53,95s
Total	154/166		153/166		149/166	

Tableau 10.5 – Nombre d'optima trouvés et temps moyens de calcul des optima

## 10.5 Conclusion

Dans ce chapitre, nous avons présenté une recherche à divergence limitée qui est une méthode de résolution approchée pour le problème de speed-dating. Nous avons proposé différentes solutions pour élaguer l'arbre de recherche et ainsi converger vers une solution optimale rapidement. Nous avons mis en évidence l'efficacité de coupes de symétries et de recherche de borne supérieures au cours de la recherche. La méthode est compétitive avec la PLNE pour certaines instances carrées. Cependant, la LDS montre ses limites pour certaines instances parmi les plus grandes et explosent en mémoire avant la fin de la recherche. Nous envisageons d'améliorer cette méthode de plusieurs manières. Dans un premier temps, il nous semble important de reconsidérer les structures de données du code pour alléger les structures de données et corriger les problèmes de taille mémoire. Ensuite, nous souhaitons définir des zones de symétrie plus grande afin de couper plus largement l'arbre de recherche. Par exemple, il est possible de casser des symétries par rapport aux dates de rencontres (cf. règle de dominance 9.3.1). Il est possible d'améliorer les coupes d'ordonnancements non maximale-actifs en les détectant plus tôt. Également, nous voulons étudier et éventuellement modifier l'ordre d'exploration de l'arbre dans un but de limiter la taille maximale de la file de nœuds. Enfin, le schéma de branchement que nous avons utilisé est en série, il est envisageable d'utiliser un autre schéma de branchement.

*Les travaux de ce chapitre ont été présentés lors de la conférence ROADEF 14 (Le Roux et al., 2014).*



# Conclusions et Perspectives

Dans ces travaux, nous nous sommes intéressés à un cas particulier de problème d’ordonnancement de rendez-vous en tête-à-tête. Ce problème particulier n’ayant pas encore été posé dans la littérature, la première partie de cette thèse a consisté à définir clairement le cadre des problèmes d’ordonnancement de rendez-vous en tête-à-tête et à proposer des notations. Nous avons étendu les notations classiques d’ordonnancement  $\alpha|\beta|\gamma$  (Graham et al., 1979). Dans le chapitre État de l’art, nous avons référencé plusieurs applications proches étudiées dans la littérature et positionné ces problèmes en regard de notre classification. Nous nous sommes également intéressés à des problèmes classiques d’ordonnancement ou de graphes ayant des similitudes avec le problème de speed-dating. De cette étude, nous avons déduit la nécessité d’une approche spécifique pour notre problème qui est nouveau. Il n’existe pas d’instances dans la littérature pour notre problème, nous avons donc décidé de générer aléatoirement un ensemble de 300 instances de 14 à 70 participants d’après des paramètres réalistes. Ces instances permettent de tester les différentes méthodes de résolution que nous développons par la suite.

Le problème que nous étudions étant nouveau, il semble essentiel de faire une étude préalable de sa complexité. Cela permet de positionner la difficulté du problème et d’orienter les méthodes de recherche à développer. Nous avons ainsi démontré que les cas sans rencontres interdites et avec retard pour au plus une population sont polynomiaux. L’un de ces principaux résultats est l’adaptation d’un algorithme polynomial dédié à un problème d’open shop dont l’objectif est le plus grand retard  $L_{\max}$ . Ensuite, nous avons démontré que l’ensemble des cas avec rencontres interdites sont  $\mathcal{NP}$ -difficiles au sens fort. Nous sommes partis d’un problème d’extension de coloration des arêtes d’un graphe biparti pour effectuer une réduction vers le cas avec rencontres interdites et retards possibles pour tous les participants pour démontrer la  $\mathcal{NP}$ -difficulté de celui-ci. Par réductions successives du cas général, nous sommes parvenus aux cas sans retards pour une population puis pour les deux populations.

Ensuite, nous avons développé des bornes inférieures et des bornes supérieures afin d’obtenir rapidement une première évaluation de l’objectif et des solutions. Les bornes inférieures sont obtenues principalement par relaxation des contraintes de disjonction des rencontres des participants. Nous affinons la qualité des bornes inférieures en relâchant le moins possible de contraintes. On arrive à un très bon compromis entre la perte de structure due à la relaxation de contraintes et la rapidité de calcul des bornes inférieures. Nous proposons également des heuristiques de type glouton qui fournissent des bornes supérieures et de premières solutions valides. Les heuristiques développées se basent sur le tri des rencontres par ordre de priorité. Deux approches sont utilisées ensuite pour ordonnancer les rencontres. La première se fait en série : à chaque date, on choisit la première rencontre dans l’ordre de priorité pour l’ordonnancer à cette date. La deuxième se fait en parallèle : à chaque date on détermine le nombre maximal de rencontres ordonnancables en optimisant la priorité totale des rencontres. La combinaison des heuristiques et bornes inférieures que nous avons développées permet de résoudre environ 44% des instances que nous avons générées à l’optimum. Ces solutions et bornes inférieures sont réutilisées dans l’ensemble des méthodes de résolution que nous avons mises au point dans la suite de ces travaux comme solutions de départ et critères d’arrêt respectivement.

La quatrième partie de ce manuscrit est consacrée à la modélisation et à la résolution du problème de speed-dating. Les premiers modèles que nous avons décrits sont des programmes linéaires en nombres entiers. Nous nous sommes efforcés d’écrire des modèles dont la relaxation linéaire est la plus petite possible par l’ajout d’inégalités valides pour renforcer les modèles. Nous avons également défini des règles de dominance limitant l’espace de recherche des solutions. Nous avons repris certaines règles classiques (ordonnancements actifs) et nous avons cassé des symétries entre participants. La PLNE fournit de bons résultats pour les petites instances, cependant elle montre ses limites lors du



passage à l'échelle. Nous avons également développé un modèle de programmation par contraintes pour le problème de speed-dating. Cette modélisation apparaît très naturelle puisqu'elle n'utilise que des contraintes classiques "All-Different" qui ont largement été étudiées dans la littérature et pour lesquelles de très bonnes techniques de filtrage sont connues. Dans le but de réduire le domaine de recherche, nous avons ajouté également des règles de dominance à ce modèle. Nous avons également développé des stratégies de recherche dédiées pour cette méthode. Le modèle de PPC s'est au final montré moins performant et moins robuste que celui de PLNE (certaines petites instances à 14 participants ne sont pas résolues en 30 min). Il est probable que la combinatoire du problème soit trop importante et que les efforts pour limiter l'exploration de l'arbre vers des solutions symétriques ne soient pas suffisants pour le réduire de manière utile. Dans le but d'adresser des problèmes de plus grande taille, nous avons ensuite développé une méthode approchée. Notre choix s'est porté sur une recherche à divergence limitée (Limited Discrepancy Search) qui est une méthode basée sur l'exploration d'un arbre tronqué. Pour ce type de méthode, il est important de disposer d'un bon critère heuristique guidant la recherche vers une solution de qualité. Les heuristiques de type glouton que nous avons développées auparavant fournissent de bons critères que nous avons exploités dans cette méthode. Les modélisations et l'étude de la structure des instances effectuées dans la partie modélisation de cette thèse nous ont amené à constater le grand nombre de solutions symétriques. Cependant, l'ajout de contraintes a priori coupant les symétries s'est montré inefficace sur les modèles de PLNE et de PPC. C'est pourquoi nous avons décidé de gérer les symétries au cours de la recherche dans la LDS plutôt que de fixer le choix de la solution dominante avant la recherche comme cela a été fait dans la partie précédente. Nous avons ainsi défini des ordonnancements partiels équivalents qui permettent de comparer deux nœuds de même niveau au cours de la recherche et de couper avant leur création des nœuds équivalents. Les résultats ont montré que le temps passé à comparer des ordonnancements au cours de la recherche est compensé par les nombreuses coupes occasionnées dans l'arbre. De plus, nous avons intégré dans cette recherche les outils qui se sont montrés les plus efficaces dans les travaux précédents de la thèse : bornes inférieures, supérieures et filtrage de type "AllDifferent". Les premiers résultats de cette méthode sont intéressants pour les instances carrées. Les temps d'obtention de solutions optimales pour les instances à 50 et 70 participants sont nettement meilleurs que ceux obtenus par la PLNE (instances les plus difficiles à résoudre pour la PLNE). Cependant, nous pensons que cette méthode pourrait être plus performante, plusieurs perspectives d'amélioration sont envisageables. La règle de dominance d'ordonnement maximalement actif peut être démontrée et son implémentation peut être améliorée par des filtrages effectués en amont. Nous pensons que les zones de symétries définissant des classes d'équivalence peuvent être étendues pour augmenter encore le nombre de branches équivalentes coupées notamment en y intégrant la règle de dominance de symétries par rapport aux dates. Nous envisageons d'alléger les structures de données du code qui pourraient être à l'origine des problèmes de mémoire qui se produisent pour certaines instances.

Les perspectives de notre travail sont nombreuses. Nous avons déjà abordé des pistes d'amélioration de la méthode de recherche à divergence limitée.

Nous souhaitons également clore le dernier cas de complexité (sans rencontres interdites et avec retards pour l'ensemble des participants). Nous pensons commencer par le problème de décision  $W_{\max} = 0$  pour lequel nous avons trouvé des propriétés intéressantes avant de généraliser au cas  $W_{\max} \leq k$ .

Des pistes sont encore à explorer pour améliorer le modèle de programmation par contraintes. Notamment, d'autres stratégies de recherche et d'autres règles de dominance peuvent être implémentées.

Nous voudrions ensuite étudier le problème avec ajout de contraintes limitant les déplacements des participants pour faciliter l'organisation de la soirée au décideur et des contraintes limitant le nombre maximal d'attentes consécutives des participants pour satisfaire ceux-ci au mieux. Nous envisageons également l'ajout d'un deuxième objectif en lexicographique de minimisation de la somme des créneaux des participants pour améliorer les solutions trouvées.

Enfin, afin de se rapprocher au mieux du cas réel, nous souhaiterions adapter nos méthodes au cas dynamique (dans lequel les dates d'arrivée des participants sont connues au fur et à mesure de leur arrivée).

# **Appendices**



## Annexe A

# Modèles de génération de colonnes

### A.1 Modèle indicé par les dates de rencontres

Le modèle suivant est une décomposition de Dantzig-Wolfe du modèle linéaire indexé sur le temps.

#### A.1.1 Problème maître

On note  $\mathcal{Meet} := \{(i, j) \in \mathcal{M} \times \mathcal{F} \mid p_{ij} = 1\}$  l'ensemble des rencontres obligatoires. Nous définissons ici les variables nécessaires à l'écriture de notre problème. On note  $\Omega$  l'ensemble des parties de  $\mathcal{Meet}$  ordonnables en même temps, c'est-à-dire :

$$\Omega = \{\omega \subset \mathcal{Meet} \mid (i, j) \in \omega, (i', j') \in \omega, i = i' \iff j = j'\}.$$

Pour  $\omega \in \Omega$  et  $d \in \mathcal{S}$  (ensemble des dates de rencontres de l'événement) :

- $x_\omega^d := \begin{cases} 1 & \text{si les rencontres ayant lieu à la date } d \text{ sont les rencontres de } \omega, \\ & \text{i.e. } \omega = \{(i, j) \in \Omega \mid C_{ij} = d + 1\}; \\ 0 & \text{sinon.} \end{cases}$
- $W_{\max}$  est le plus grand nombre de créneaux d'attentes parmi les participants.

L'écriture du modèle nécessite également de définir les constantes : pour  $(i, j) \in \mathcal{Meet}, \omega \in \Omega$  et  $d \in \mathcal{S}$

- $\delta_{ij}^\omega := \begin{cases} 1 & \text{si } (i, j) \in \omega; \\ 0 & \text{sinon.} \end{cases}$

Le problème maître s'écrit de la manière suivante.

$$\min W_{\max} \tag{A.1}$$

s.c.

$$\forall (i, j) \in \mathcal{Meet}, \quad W_{\max} - \sum_{d \geq r_{ij}} \sum_{\omega \in \Omega} (d + 1) \delta_{ij}^\omega x_\omega^d \geq -d_{ij} \quad (\alpha_{ij} \geq 0) \tag{A.2}$$

$$\forall (i, j) \in \mathcal{Meet}, \quad \sum_{d \geq r_{ij}} \sum_{\omega \in \Omega} \delta_{ij}^\omega x_\omega^d = 1 \quad (\beta_{ij}) \tag{A.3}$$

$$\forall d \in \mathcal{S}, \quad \sum_{\omega \in \Omega} x_\omega^d \leq 1 \quad (\gamma_d \leq 0) \tag{A.4}$$

$$\forall \omega \in \Omega, \forall d \in \mathcal{S}, \quad x_\omega^d \geq 0. \tag{A.5}$$

Les contraintes (A.2) effectuent le lien entre les variables binaires  $x_{\omega}^d$  et l'objectif. Si la rencontre  $(i, j)$  a lieu à la date  $d$ , alors l'objectif  $W_{\max}$  est supérieur ou égal à  $d + 1 - d_{ij}$ . Les contraintes (A.3) assurent que chaque rencontre (non interdite) a lieu une et une seule fois lors de la soirée, en particulier, les  $\omega$  retenus forment une partition de l'ensemble des rencontres  $\mathcal{Meet}$ . Les contraintes (A.4) assurent que deux sous-ensembles de rencontres différents  $\omega_1$  et  $\omega_2$  ne sont pas ordonnancés à la même date.

À chaque contrainte est associée un multiplicateur lagrangien, notés respectivement  $\alpha_{ij}$ ,  $\beta_{ij}$  et  $\gamma_d$  pour les contraintes (A.2), (A.3) et (A.4).

### A.1.2 Sous-problèmes

Le sous-problème consiste à déterminer un ensemble de rencontres et la date  $d$  à laquelle les ordonnancer. On détermine pour chaque date  $d \in \mathcal{S}$  s'il existe un sous-ensemble  $\omega$  de rencontres telles que le coût réduit de la variable  $x_{\omega}^d$  est négatif.

On définit ici les variables du sous-problème. Pour  $(i, j) \in \mathcal{Meet}$  et  $\omega \in \Omega$ ,

$$\bullet \delta_{ij} = \begin{cases} 1 & \text{si } (i, j) \in \omega, \\ 0 & \text{sinon.} \end{cases}$$

Le sous-problème pour une date  $d \in \mathcal{S}$  fixée est le suivant :

$$\min \sum_{(i,j) \in \mathcal{Meet}} \alpha_{ij}(d+1)\delta_{ij} - \sum_{(i,j) \in \mathcal{Meet}} \beta_{ij}\delta_{ij} - \gamma_d \quad (\text{A.6})$$

s.c.

$$\forall M_i \in \mathcal{M}, \quad \sum_{F_j \in \mathcal{F}, p_{ij}=1} \delta_{ij} \leq 1 \quad (\text{A.7})$$

$$\forall F_j \in \mathcal{F}, \quad \sum_{M_i \in \mathcal{M}, p_{ij}=1} \delta_{ij} \leq 1 \quad (\text{A.8})$$

$$\forall (i, j) \in \mathcal{Meet} \text{ tel que } d < r_{ij}, \quad \delta_{ij} = 0 \quad (\text{A.9})$$

$$\forall (i, j) \in \mathcal{Meet}, \quad \delta_{ij} \in \{0, 1\} \quad (\text{A.10})$$

Les contraintes (A.7) et (A.8) sont des contraintes de non ubiquité des participants. Enfin, les contraintes (A.9) interdisent l'ordonnancement à la date  $d$  de participants n'étant pas encore arrivés à cette date.

Il s'agit d'un problème d'affectation (problème de couplage maximum à coût minimum dans un graphe biparti).

*Remarque A.1.1.* La matrice de contraintes de ce sous-problème est totalement unimodulaire. En effet, cette matrice est la matrice d'incidence nœud-arête du graphe biparti des rencontres. Chaque ligne représente un sommet de  $U \cup V$  et chaque colonne représente une arête du graphe, c'est-à-dire une rencontre. Une entrée de la matrice est égale à 1 si et seulement si le sommet est incident à une arête du graphe (le participant est l'un des deux participants de la rencontre). Une matrice d'incidence nœud-arête du graphe biparti des rencontres est totalement unimodulaire (Nemhauser et Wolsey, 1988), chapitre III.1 Integral Polyhedra.

## A.2 Modèle indicé par les participantes

Les colonnes du modèle présenté dans cette section représentent l'emploi du temps de chaque femme.

### A.2.1 Problème maître

On note  $\mathcal{L} = \{l = (l_0, l_1, \dots, l_{m-1}) \in (\{-1\} \cup \mathcal{S})^n \mid 0 \leq i_1, i_2 \leq m-1, i_1 \neq i_2 \Rightarrow l_{i_1} \neq l_{i_2}\}$ , l'ensemble des listes de dates qu'on peut affecter à une participante. Si on affecte la liste  $l \in \mathcal{L}$  à la femme  $F_j \in \mathcal{F}$ , cette participante

rencontre l'homme  $M_i \in \mathcal{M}$  à la date  $l_i$ ; si la rencontre est interdite, on force  $l_i = -1$ . Pour chaque femme  $F_j \in \mathcal{F}$ , toutes les listes de  $\mathcal{L}$ , ne sont pas disponibles. On note l'ensemble des listes de dates disponibles pour la femme  $F_j$  :

$$\mathcal{L}_j = \{l \in \mathcal{L} \mid \forall i \in \mathcal{M}, \text{ si } p_{ij} = 0 \text{ alors } l_i = -1, \text{ sinon } l_i \geq r_{ij}\}.$$

On utilise les variables suivantes. Pour  $j \in \mathcal{F}, l \in \mathcal{L}_j$  :

$$\bullet x_l^j = \begin{cases} 1 & \text{si la liste de dates de rencontres de la femme } F_j \text{ est } l, \\ 0 & \text{sinon.} \end{cases}$$

On rappelle le symbole de Kronecker, pour deux nombres  $x$  et  $y$ ,  $\delta_x^y = 1$  si  $x = y$  et 0 sinon.

Le problème maître s'écrit de la manière suivante.

$$\min W_{\max} \tag{A.11}$$

s.c.

$$\forall (i, j) \in \text{Meet}, \quad W_{\max} - \sum_{l \in \mathcal{L}_j} \sum_{d \geq r_{ij}} (d+1) \delta_d^{l_i} x_l^j \geq -d_{ij} \quad (\mu_{ij} \geq 0) \tag{A.12}$$

$$\forall M_i \in \mathcal{M}, \forall d \in \mathcal{S}, \quad \sum_{\substack{j \in \mathcal{F} \\ p_{ij}=1}} \sum_{l \in \mathcal{L}_j} \delta_d^{l_i} x_l^j \leq 1 \quad (\nu_{id} \leq 0) \tag{A.13}$$

$$\forall F_j \in \mathcal{F}, \quad \sum_{l \in \mathcal{L}_j} x_l^j = 1 \quad (\pi_j) \tag{A.14}$$

$$\forall F_j \in \mathcal{F}, \forall l \in \mathcal{L}_j, \quad x_l^j \geq 0. \tag{A.15}$$

Les contraintes (A.12) effectuent le lien entre les variables binaires  $x_l^j$  et l'objectif  $W_{\max}$ . Les contraintes (A.13) interdisent qu'un homme fasse plusieurs rencontres pendant le même créneau. Les contraintes (A.14) forcent le choix d'une unique liste de rencontres pour chaque femme. Les contraintes d'intégrité des variables sont relâchées dans (A.15).

### A.2.2 Sous-problèmes

Le sous-problème associé au problème maître précédent consiste à déterminer une femme et une liste de dates de rencontres pour cette participante.

Les variables des sous-problèmes sont les suivantes.

$$\bullet \delta_d^{l_i} = \begin{cases} 1 & \text{si } l_i = d; \\ 0 & \text{sinon.} \end{cases}$$

Le sous-problème pour la femme  $F_j \in \mathcal{F}$  s'écrit de la manière suivante.

$$\min \sum_{(i,j) \in \text{Meet}} \sum_{d=r_{ij}}^{s-1} (d+1) \mu_{ij} \delta_d^{l_i} - \sum_{\substack{i \in \mathcal{M} \\ p_{ij}=1}} \sum_{d=r_{ij}}^{s-1} \nu_{id} \delta_d^{l_i} - \pi_j \tag{A.16}$$

s.c.

$$\forall d \in \llbracket r_j^{(2)}, s-1 \rrbracket, \quad \sum_{\substack{i \in \mathcal{M} \\ p_{ij}=1}} \delta_d^{l_i} \leq 1 \tag{A.17}$$

$$\forall M_i \in \mathcal{M}, \text{ tel que } p_{ij} = 1, \quad \sum_{d=r_{ij}}^{s-1} \delta_d^{l_i} = 1 \tag{A.18}$$

$$\forall M_i \in \mathcal{M}, \forall d \in \mathcal{S}, \quad \delta_d^{l_i} \in \{0, 1\}. \tag{A.19}$$

La fonction objectif (A.16) vise à minimiser le coût réduit de la nouvelle variable entrant dans le problème maître restreint. Les contraintes (A.17) sont des contraintes de non-ubiquité pour la femme  $F_j$ . Les contraintes (A.18) déterminent une unique date de rencontre pour tous les hommes devant rencontrer  $F_j$ .

*Remarque A.2.1.* La matrice de contraintes de ce sous-problème est également totalement unimodulaire car il s'agit de la matrice d'incidence nœud-arête d'un graphe biparti. Les sommets de ce graphe sont d'une part les hommes que doit rencontrer la femme  $F_j$  et d'autre part les dates durant lesquelles  $F_j$  est disponible. Une arête de ce graphe relie un homme à une date si celui-ci est disponible à cette date.

# Bibliographie

- Artigues, C., Lopez, P., et Ayache, P.-D. (2005). Schedule generation schemes for the job-shop problem with sequence-dependent setup times : Dominance properties and computational analysis. *Annals of Operations Research*, 138(1) :21–52. [69](#)
- Asratian, A. S. et Casselgren, C. (2007). On interval edge colorings of  $(\alpha, \beta)$ -biregular bipartite graphs. *Discrete Mathematics*, 307(15) :1951 – 1956. [26](#)
- Baker, K. R. (1974). *Introduction to sequencing and scheduling*, volume 15. Wiley New York. [88](#)
- Bartholdi, J. J. et McCroan, K. L. (1990). Scheduling interviews for a job fair. *Operations Research*, 38(6) :951–960. [22](#)
- Beldiceanu, N., Carlsson, M., et Rampon, J.-X. (2005). Global constraint catalog. *SICS Research Report*. [100](#), [101](#)
- Blazewicz, J., Dror, M., et Weglarz, J. (1991). Mathematical programming formulations for machine scheduling : A survey. *European Journal of Operational Research*, 51(3) :283 – 300. [83](#)
- Bofill, M., Espasa, J., Garcia, M., Palahí, M., Suy, J., et Villaret, M. (2014). Scheduling b2b meetings. In O’Sullivan, B., editor, *Principles and Practice of Constraint Programming*, volume 8656 of *Lecture Notes in Computer Science*, pages 781–796. Springer International Publishing. [23](#)
- Boudhar, M. et Finke, G. (2000). Scheduling on a batch machine with job compatibilities. *Belgian Journal of Operations Research*, 40:69–80. [25](#)
- Bräsel, H., Tautenhahn, T., et Werner, F. (1993). Constructive heuristic algorithms for the open shop problem. *Computing*, 51(2) :95–110. [69](#)
- Brucker, P. (2001). *Scheduling Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition. [57](#), [59](#), [88](#)
- Choco-Team (2010). Choco : an open source java constraint programming library. *École des Mines de Nantes, Research report*. [103](#)
- Chudnovsky, M., Robertson, N., Seymour, P., et Thomas, R. (2006). The strong perfect graph theorem. *Annals of Mathematics*, 164 :51–229. [42](#)
- Colbourn, C. (1984). The complexity of completing partial latin squares. *Discrete Applied Mathematics*, 8(1) :25–30. [44](#)
- Colbourn, C. J. et Nonay, G. (1997). A golf design of order 11. *Journal of Statistical Planning and Inference*, 58(1) :29 – 31. [24](#)
- Cole, R. et Hopcroft, J. (1982). On edge coloring bipartite graphs. *SIAM Journal on Computing*, 11(3) :540–546. [40](#)
- de Werra, D. (1988). Some models of graphs for scheduling sports competitions. *Discrete Applied Mathematics*, 21(1) :47 – 65. [24](#), [79](#)
- Easton, K., Nemhauser, G., et Trick, M. (2001). The traveling tournament problem description and benchmarks. In Walsh, T., editor, *Principles and Practice of Constraint Programming — CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 580–584. Springer Berlin Heidelberg. [24](#)



- Edmonds, J. et Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2) :248–264. [71](#)
- Ernst, A. T., Mills, R. G. J., et Welgama, P. (2003). Scheduling appointments at trade events for the australian tourist commission. *Interfaces*, 33(3) :12–23. [22](#)
- Even, S., Itai, A., et Shamir, A. (1976). On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4) :691–703. [23](#), [25](#), [26](#), [79](#)
- Fernández-Baca, D. et Martel, C. (1989). On the efficiency of maximum-flow algorithms on networks with small integer capacities. *Algorithmica*, 4(1-4) :173–189. [39](#)
- Gandibleux, X., Pajot, B., et Poncelet, D. (2006). Modèle en variables 0–1 et système d’aide pour l’optimisation de l’organisation de bourses d’échanges de technologies. In *ROADEF 06*, Lille. [22](#)
- Garey, M. R. et Johnson, D. S. (1979). *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York, NY, USA. [52](#)
- Gent, I. P., Petrie, K. E., et Puget, J.-F. (2006). Chapter 10 - symmetry in constraint programming. In Francesca Rossi, P. v. B. et Walsh, T., editors, *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*, pages 329 – 376. Elsevier. [101](#)
- Gharbi, A. et Mahjoubi, A. (2013). New lower bounds for flow shop scheduling. *International Journal of Humanities and Management Sciences (IJHMS)*, 1 :125. [58](#)
- Graham, R. L., Lawler, E. L., Lenstra, J. K., et Rinnooy Kan, A. H. G. (1979). Optimization And Approximation In Deterministic Sequencing And Scheduling : A Survey. *Annals of Discrete Mathematics*, 5 :287–326. [17](#), [119](#)
- Grötschel, M., Lovász, L., et Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1 :169–197. [42](#)
- Guéret, C., Bellenguez-Morineau, O., Pavageau, C., Péton, O., et Poncelet, D. (2009). Optimisation de la planification de bourses d’échanges de technologies. In *ROADEF 09*, pages 177–178, Nancy. [22](#)
- Guéret, C. et Prins, C. (1999). A new lower bound for the open shop problem. *Annals of Operations Research*, 92(0) :165–183. [57](#)
- Harvey, W. D. et Ginsberg, M. L. (1995). Limited discrepancy search. In *Proceedings IJCAI’95*, pages 607–613. Morgan Kaufmann. [107](#)
- Haupt, R. (1989). A survey of priority rule-based scheduling. *Operations-Research-Spektrum*, 11(1) :3–16. [69](#)
- Hopcroft, J. et Karp, R. (1973). An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4) :225–231. [65](#)
- Huang, Y., Zhang, X., et Alexander, P. S. (2012). A heuristic algorithm for optimizing business matchmaking scheduling. *International Journal of Operations Research and Information Systems*, 3(4) :59–73. [22](#)
- Jouglet, A. et Carlier, J. (2011). Dominance rules in combinatorial optimization problems. *European Journal of Operational Research*, 212(3) :433 – 444. [84](#)
- Katz, B., Rutter, I., Strasser, B., et Wagner, D. (2011). Speed dating. In Pardalos, P. et Rebennack, S., editors, *Experimental Algorithms*, volume 6630 of *Lecture Notes in Computer Science*, pages 292–303. Springer Berlin Heidelberg. [23](#)
- Kellerer, H., Tautenhahn, T., et Woeginger, G. (1995). Note : Open-shop scheduling with release dates to minimize maximum lateness. *Naval Research Logistics (NRL)*, 42(1) :141–145. [24](#), [36](#), [37](#), [39](#), [41](#)
- König, D. (1916). über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77(4) :453–465. [22](#), [40](#)

- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2) :83–97. [71](#)
- Lapègue, T., Bellenguez-Morineau, O., et Guéret, C. (2011). Scheduling of a speed-dating event. In *OR 2011*, pages 99–100. [24](#), [100](#)
- Lasdon, L. S. (1970). *Optimization theory for large systems*. Courier Dover Publications. [97](#)
- Le Pape, C. et Baptiste, P. (1999). Heuristic control of a constraint-based algorithm for the preemptive job-shop scheduling problem. *Journal of Heuristics*, 5(3) :305–325. [107](#)
- Le Roux, A., Bellenguez-Morineau, O., et Guéret, C. (2013a). Bounds and valid inequalities for speed-dating linear models. In *MAPSP 2013*, Pont-à-Mousson. [76](#), [98](#)
- Le Roux, A., Bellenguez-Morineau, O., et Guéret, C. (2013b). Résolution d'un problème de speed-dating par programmation linéaire en nombres entiers. In *ROADEF 13*, Troyes. [98](#)
- Le Roux, A., Bellenguez-Morineau, O., et Guéret, C. (2014). Limited discrepancy search pour le problème d'ordonnement de rendez-vous. In *ROADEF 14*, Bordeaux. [117](#)
- Le Roux, A., Bellenguez-Morineau, O., Guéret, C., et Prot, D. (2012). Complexité de problèmes d'ordonnement de rencontres de type speed-dating. In *ROADEF 12*, Angers. [53](#)
- Lipski, W. et Preparata, F. P. (1981). Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems. *Acta Informatica*, 15 :329–346. [63](#)
- Malapert, A., Cambazard, H., Guéret, C., Jussien, N., Langevin, A., et Rousseau, L.-M. (2012). An optimal constraint programming approach to the open-shop problem. *INFORMS Journal on Computing*, 24(2) :228–244. [99](#)
- Marchand, H. et Wolsey, L. A. (1999). The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming*, 85 :15–33. [81](#)
- Minoux, M. (2007). *Programmation mathématique : théorie et algorithmes*, 2e édition. Ted & Doc - Lavoisier. [97](#)
- Nemhauser, G. L. et Wolsey, L. A. (1988). *Integer and combinatorial optimization*, volume 18. Wiley New York. [98](#), [124](#)
- Pesant, G. (2004). A regular language membership constraint for finite sequences of variables. In Wallace, M., editor, *Principles and Practice of Constraint Programming – CP 2004*, volume 3258 of *Lecture Notes in Computer Science*, pages 482–495. Springer Berlin Heidelberg. [103](#)
- Pinedo, M. L. (2012). *Scheduling : theory, algorithms, and systems*. Springer. [83](#)
- Puget, J.-F. (1993). On the satisfiability of symmetrical constrained satisfaction problems. In Komorowski, J. et Raś, Z., editors, *Methodologies for Intelligent Systems*, volume 689 of *Lecture Notes in Computer Science*, pages 350–361. Springer Berlin Heidelberg. [101](#)
- Régin, J. C. (1994). A filtering algorithm for constraints of difference in cps. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, pages 362–367. [110](#), [115](#)
- Régin, J.-C. (2011). Global constraints : A survey. In van Hentenryck, P. et Milano, M., editors, *Hybrid Optimization*, volume 45 of *Springer Optimization and Its Applications*, pages 63–134. Springer New York. [100](#)
- Rinaldi, F. et Serafini, P. (2007). Scheduling school meetings. In Burke, E. et Rudová, H., editors, *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, pages 280–293. Springer Berlin Heidelberg. [21](#)
- Sen, S. et Durfee, E. (1998). A formal study of distributed meeting scheduling. *Group Decision and Negotiation*, 7(3) :265–289. [24](#)

- Sevastjanov, S. (1990). Interval colorability of the edges of a bipartite graph. *Metody Diskretnogo Analiza*, 50:61–72. [26](#)
- Steiner, G. et Yeomans, J. (1996). A linear time algorithm for maximum matchings in convex, bipartite graphs. *Computers & Mathematics with Applications*, 31(12) :91–96. [63](#)
- Tassa, T. (2012). Finding all maximally-matchable edges in a bipartite graph. *Theoretical Computer Science*, 423(0) :50 – 58. [110](#), [115](#)
- Tassa, T. (2013). Addendum to “finding all maximally-matchable edges in a bipartite graph” [theoret. comput. sci. 423 (2012) 50–58]. *Theoretical Computer Science*, 491(0) :136 –. [110](#), [115](#)
- Valouxis, C. et Housos, E. (2003). Constraint programming approach for school timetabling. *Computers & OR*, 30(10) :1555–1572. [99](#)
- Walsh, T. (2006). General symmetry breaking constraints. In Benhamou, F., editor, *Principles and Practice of Constraint Programming - CP 2006*, volume 4204 of *Lecture Notes in Computer Science*, pages 650–664. Springer Berlin Heidelberg. [101](#)
- Yoshikawa, M., Kaneko, K., Yamanouchi, T., et Watanabe, M. (1996). A constraint-based high school scheduling system. *IEEE Intelligent Systems*, 11(1) :63–72. [99](#)



# Thèse de Doctorat

**Agnès LE ROUX**

**Ordonnancement de rendez-vous en tête-à-tête**

**One-to-one meeting scheduling**

## Résumé

Les problèmes d'ordonnancement de rendez-vous en tête-à-tête sont des problèmes dans lesquels des personnes souhaitent se rencontrer par deux lors de courts rendez-vous qui se déroulent lors d'une session unique. Dans cette thèse, nous référençons plusieurs applications de ce type de problèmes et proposons des notations qui généralisent les notations standards de problèmes d'ordonnancement  $\alpha|\beta|\gamma$ . Nous nous intéressons en particulier à un cas dans lequel deux populations distinctes se rencontrent, des participants peuvent arriver en retard et des rencontres sont interdites. L'objectif est de minimiser le nombre maximal d'attentes des participants. Nous étudions dans un premier temps la complexité de ces problèmes : nous démontrons que plusieurs cas sans rencontre interdite sont polynomiaux et que le cas général est  $\mathcal{NP}$ -complet au sens fort. Nous proposons ensuite des bornes inférieures. Puis nous développons plusieurs méthodes de résolution. Des modèles de programmation linéaire en nombres entiers et un modèle de programmation par contraintes sont tout d'abord proposés. Des règles de dominance permettant de limiter les symétries sont intégrées à ces modèles dans le but de limiter l'espace des solutions. Enfin, nous proposons une recherche à divergence limitée (limited discrepancy search) qui est une méthode approchée basée sur l'exploration d'un arbre de recherche tronqué. Dans cette méthode, nous exploitons le plus possible les propriétés de symétrie du problème pour faciliter la convergence vers une bonne solution. Toutes ces méthodes sont testées et comparées sur un ensemble de 300 instances générées aléatoirement d'après des paramètres réalistes.

## Mots clés

ordonnancement, rendez-vous en tête-à-tête, complexité algorithmique, coloration d'arêtes d'un graphe, programmation linéaire en nombres entiers, programmation par contraintes, limited discrepancy search.

## Abstract

One-to-one meeting scheduling problems are problems where a population of actors want to meet each other during short time slots that take place in a single session. In this thesis, we reference several applications of this type of problems found in the literature and introduce a notation extending the well-known scheduling notation  $\alpha|\beta|\gamma$ . We are particularly interested in a case in which two distinct populations meet, participants may arrive late and some meetings are forbidden. The objective is to minimize the maximum number of participants waiting slots. First, we study the complexity of these problems: we show that several cases with no forbidden meeting are polynomial and that the general case is NP-complete in the strong sense. We then propose lower bounds. After that, we develop several resolution methods. Integer linear programming models and a constraint programming model are developed. To limit the solution space, we add dominance rules based on symmetries to these methods. Finally, we present a limited discrepancy search (i.e. an approximate method based on the exploration of a truncated tree search). In this method, we use as much as possible the symmetry properties of the problem to facilitate the convergence to a good solution. All these methods are tested and compared on a set of 300 randomly generated instances from realistic parameters.

## Key Words

scheduling, one-to-one meetings, computational complexity, graph edge-coloring, integer linear programming, constraints programming, limited discrepancy search.

